Budget-Constrained Shortest Paths in Continuous Domains

Michal Lazar[1] and Noa Agmon[1]

Department of Computer Science, Bar-Ilan University, Ramat Gan, Israel
michallazar88@gmail.com, agmon@cs.biu.ac.il

Abstract. Autonomous robots operating in continuous environments must often navigate through regions that consume limited resources, such as fuel, energy, or risk exposure, while minimizing travel distance. Referring to those special regions as *budget zones*, we formalize the **Circular Budget-Constrained Shortest Path (C-BCSP)** problem: Find the shortest Euclidean path from source to target in a plane containing n disjoint circular budget zones, where entering zone c_i incurs exactly b_i units of a global budget B.
Proving that C-BCSP is NP-Hard, we introduce the Iterative Refinement Discretization (IRD) algorithm, based on a bounded deviation theorem showing that the continuous optimal path lies within a narrow corridor around any discretized solution. This insight allows a significant reduction in the search space, while provably converging to the optimal continuous solution. We empirically compare IRD with a fully discretized visibility-graph baseline (VG-RCSP) and a budget-constrained RRT* variant (CC-RRT*) and show that while all methods achieve comparable path quality, IRD reduces runtime by over an order of magnitude and maintains significantly smaller graph sizes. Finally, we show that our theoretical results and the IRD algorithm guarantees extend naturally to spherical budget zones in $\mathbb{R}^3$.

1 INTRODUCTION

Autonomous robots operating in continuous environments frequently encounter regions that are traversable but impose resource costs, such as fuel in difficult terrain, energy for specialized locomotion, or risk exposure in hazardous areas. Our formulation treats b_i as an indivisible per-zone commitment — either skip c_i or consume the full b_i — matching scenarios such as a full surveillance or inspection sweep, a minimum-safe hazard crossing, or a mandatory refuel stop, where partial entry has no utility; under a constant traversal speed this equivalently captures time or coverage budgets. Unlike traditional obstacles, these *budget zones* do not block passage; rather, entering zone c_i incurs a fixed cost b_i against a limited global budget B. Planning an optimal path thus requires jointly deciding which zones to enter and how to route through them, minimizing travel distance while satisfying budget constraints. Following established models for budget-constrained planning in continuous space, we adopt circular budget zones. [12–14].

This continuous formulation distinguishes our problem from classical shortest-path planning. The weighted region problem [3,11], radar exposure models, and path planning in environments containing threat zones [12–14] assign distance-proportional costs to regions, where the cost accumulates continuously as a function of the path length traversed within a region. In contrast, our formulation imposes a different cost structure: budget consumption is *zero* outside budget zones, and if zone c_i is entered, the path must traverse exactly b_i units inside it. Graph-based variants such as the weight-constrained shortest path problem [5] are NP-hard, but operate on finite edge sets. The continuous setting introduces additional complexity: zone boundaries admit infinitely many candidate entry and exit points, making naive discretization computationally prohibitive.

We formalize C-BCSP and prove its NP-hardness via reduction from Subset Sum. We develop structural results showing optimal paths remain within a composed convex hull, enabling search space reduction. Our main contribution is the *Iterative Refinement Discretization (IRD)* algorithm, based on a *bounded deviation theorem*: any discretized optimal path lies within arc-length $d/2$ of the continuous optimal path, where d is the discretization step. This reduces search complexity from $O(k^{2n})$ to $O(3^n)$ per iteration while converging to the optimal solution. We empirically compare IRD against a fully discretized baseline and a budget-constrained RRT* variant, showing over an order-of-magnitude runtime improvement and substantially smaller graph sizes while maintaining solution quality. Finally, we extend IRD to spherical budget zones in $\mathbb{R}^3$.

Contributions. (i) C-BCSP formulation with per-zone and global budget constraints; (ii) NP-hardness via reduction from Subset Sum; (iii) structural properties for visibility-graph pruning; (iv) bounded deviation theorem and IRD with convergence guarantees; (v) empirical comparison of VG-RCSP, IRD, and CC-RRT*; (vi) extension to spherical zones in $\mathbb{R}^3$.

2 RELATED WORK

Shortest path planning has been extensively studied in environments with polygonal and curved obstacles. In continuous settings, optimal paths are commonly computed by discretizing the space using visibility or tangent graphs, Voronoi diagrams, grids, or sampling-based planners such as PRM, RRT, and asymptotically optimal variants like BIT* and AORRTC [7–9,17,18]. These methods typically assume that obstacles are not traversable [2]. Unlike traditional sampling-based approaches, our method does not rely on uniform sampling; instead, it employs adaptive refinement strategies, focusing on promising entry and exit points and using problem-specific prefiltering to reduce the search space without compromising optimality.

Several related problems allow traversal through special regions at additional cost. The weighted region problem [3, 11] and the bichromatic bicriteria path problem [15] assign distance-dependent penalties to regions and bound traversal within them. Radar exposure minimization and UAV threat-aware planning [12–14] similarly assume continuous risk accumulation. In contrast, our formulation

imposes *event-based* budget consumption with mandatory in-region traversal. Upon entering a budget zone c_i, a path must traverse at least a fixed distance b_i within c_i. Traversal outside all budget zones incurs zero cost. A path is feasible only if the sum of required in-zone traversal lengths does not exceed a global budget B. Among all feasible paths, we seek the one with minimum Euclidean length.

This cost structure depends on whether and how regions are entered, rather than on continuous accumulation along the path. As a result, the problem cannot be reduced to a distance-weighted shortest-path formulation, and existing algorithms for weighted or exposure-based planning do not directly apply.

Budget constraints have also been studied in discrete graphs, most notably in the weight-constrained shortest path problem (WCSP) [5], which is NP-hard. Related formulations in continuous space include the Euclidean shortest path with minimum constraint removal [1, 6], which seeks a shortest path under a total removal budget and is NP-hard even for simple obstacles; unlike our formulation, it alters the environment by removing obstacles, whereas we allow traversal through intact budget zones subject to a global in-zone budget B and binary per-zone usage (0 or b_i). More recent frameworks that jointly couple discrete and continuous decisions in motion planning, shortest paths in graphs of convex sets [10] and bidirectional transition-based RRT for cost spaces [4], address related discrete/continuous trade-offs from optimization-based and sampling-based perspectives, respectively.

Our problem differs from related work by focusing on path planning in *continuous* space with budget zones subject to both fixed per-zone costs and a global budget, requiring the selection of which zones to traverse and how, while minimizing the total Euclidean path length.

3 PROBLEM DEFINITION

We study path planning in a continuous plane containing a set $C = \{c_1, \ldots, c_n\}$ of n disjoint circular budget zones.[1] Each circle c_i has center o_i, radius r_i, boundary (circumference) ∂c_i, and an associated per-zone budget $b_i > 0$ (if entered, the in-zone length must be exactly b_i, and travel along ∂c_i does *not* consume budget). In addition, a global budget B limits the total path length traveled inside all budget zones. We assume each b_i is *beneficial*: entering c_i and consuming b_i cannot increase the total path length, so by the triangle inequality the optimum realizes b_i via a single chord (shown later).

Given source s and target t, let P_{st} denote a continuous path from s to t with length $\ell(P_{st})$. A path P_{st} is **B-legal** if: (i) $\forall i, \ell(P_{st} \cap c_i) \in \{0, b_i\}$ (per-zone feasibility); (ii) $\sum_i \ell(P_{st} \cap c_i) \leq B$ (global feasibility).

The **Circular Budget-Constrained Shortest Path (C-BCSP)** problem is to find a shortest B-legal path from s to t.

[1] We present the setting in an obstacle-free workspace; extending to polygonal obstacles is straightforward by incorporating standard visibility-graph.

4 COMPUTATIONAL HARDNESS

To prove NP-hardness of the C-BCSP problem, we consider its decision version, referred to as the Free-Space Constrained Shortest Path (FCSP), and prove its hardness via the following reduction chain:

$$\text{Subset Sum} \ \leq_p \ \text{Knapsack (decision)} \ \leq_p \ \textsf{FCSP}/\textsf{C-BCSP}.$$

Formally, in FCSP the goal is not only to minimize the total path length given the budget zones, but also to limit the path length in free space. All settings remain as defined in Section 3, with the addition of a free-space budget A, which constrains the total length of the path outside all budget zones. Showing that FCSP is NP-complete directly implies the hardness of C-BCSP.
Decision question: Does there exist a path P_{st} from s to t such that

$$\ell(P_{st} \cap (W \setminus C)) \leq A, \quad \ell(P_{st} \cap c_i) \in \{b_i, 0\} \ \forall i, \quad \ell(P_{st} \cap C) \leq B \ ?$$

Decision Knapsack Problem. Given items $i = 1, \ldots, n$ with weight $w_i > 0$ and value $v_i > 0$, capacity $W > 0$, and target $V > 0$, does there exist a subset S such that $\sum_{i \in S} w_i \leq W$ and $\sum_{i \in S} v_i \geq V$? This problem is NP-complete.

Theorem 1. *The C-BCSP decision problem is NP-hard.*

Reduction. We first recall that the standard reduction from Subset Sum to the decision version of 0–1 Knapsack allows us, without loss of generality, to assume that $v_i = 2w_i$, $W = L$, and $V = 2L$, where L is the Subset Sum target.

Given such a knapsack instance, we place s and t on the x-axis and construct an environment with n independent *gadgets* placed along $\overline{st}$. For each item i, we create three equal-radius $R_i := w_i/2$, mutually tangent circles centered on the $\overline{st}$-line (see Fig. 1) such that the left and right circles have per-zone budget 0, and the middle circle has per-zone budget $b_i := w_i$.

Let $\mathbb{N} := \{1, \ldots, n\}$ denote the full item index set, and $\mathbb{S} \subseteq \mathbb{N}$ is the subset of items selected by the path (i.e., items whose middle circles are entered). Hence the free-space budget, denoted A, equals $\sum_{i \in \mathbb{N}} w_i + \sum_{i \in \mathbb{S}} \frac{\pi}{2} w_i + \sum_{i \in \mathbb{N}} v_i + V$

We place the outer-circle tangency points so that the straight free-space traversal across the gadget (skipping the middle circle) has length v_i (here $v_i = 2w_i$). The gadget enforces the following binary choice:

1. **Skip item i.** Traverse the gadget with a straight segment of free-space length v_i; free-space budget consumed: v_i.
2. **Take item i.** Enter the middle circle once via a chord of length w_i (consuming w_i budget), and traverse the remainder between the outer circles along two quarter-circle boundary arcs (free space) of total length $(\pi/2)\, w_i$.

The left and right circles have per-zone budget 0, hence their interiors are non-traversable and act as disk obstacles, while the middle circle c_i is either skipped ($\ell(P \cap c_i) = 0$) or, if entered, must consume exactly w_i, so partial entry is infeasible. Moreover, for disjoint circular obstacles any shortest free-space path

between boundary contact points consists only of straight segments tangent to obstacle boundaries (possibly with boundary arcs), implying that when c_i is skipped the shortest traversal has length exactly v_i by construction, and the only way to obtain a strictly shorter traversal is to enter c_i, which forces consuming w_i via a chord. Since $v_i = 2w_i > (\pi/2)w_i$, taking item i strictly reduces the free-space length while consuming w_i units of inside-circle budget.

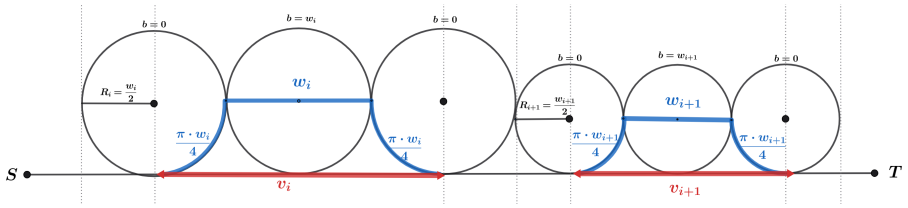

Fig. 1: Reduction gadget per item i: three tangent circles along $\overline{st}$. Outer circles (budget 0) act as obstacles; middle circle has budget $b_i{=}w_i$.

The total length of the path outside the circles will be

$$\sum_{i\in\mathbb{N}}(w_i) + \sum_{i\in\mathbb{S}}\frac{\pi}{2}(w_i) + \sum_{i\notin S} v_i \le A \Rightarrow \sum_{i\notin S} v_i \le \sum_{i\in\mathbb{N}} v_i - V \Rightarrow \sum_{i\in S} v_i \ge V$$

And inside the circles: $\sum_{i\in S} w_i < B$, so this way the $\sum_{i\in S} w_i < W$ and $\sum_{i\in S} v_i \ge V$. Thus, a feasible FCSP path exists if and only if the corresponding knapsack instance admits a feasible solution.

Finally, feasibility of a given path is verified in polynomial time by computing its intersections with the circles and measuring the corresponding Euclidean lengths; thus, FCSP $\in$ NP. Together with the reduction above, this implies that FCSP is NP-complete, and consequently C-BCSP is NP-hard.

5 EFFICIENT SEARCH STRUCTURAL PROPERTIES

We state geometric properties that underpin our algorithms. Proof sketches follow standard arguments and are omitted for brevity. Throughout, $\ell(\cdot)$ denotes Euclidean length (and $\ell(X, Y)$ the Euclidean distance between points X and Y), while $\overset{\frown}{\ell}(\cdot,\cdot)$ denotes arc length along a circle boundary. A transition point between consecutive path elements (segments, arcs, or chords) s_i and s_{i+1} is referred to as a *breaking point*, denoted by $p_{(i,i+1)}$.

Lemma 1. Boundary Breaking Points. All breaking points $p_{(i,i+1)}$ of a shortest path P^* reside on circumferences of circles.

Lemma 2. No Multiple Segments. An optimal path P^* will not contain two or more straight segments

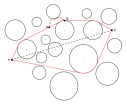

Fig. 2: st-CCH

inside a single circle; such patterns are weakly dominated by a single chord plus arcs.

Lemma 3. Order Invariant Length. Assume an optimal path P^* intersects some circle c_i such that it contains both a chord and arcs within it. $\ell(P^*)$ is the same, whether the chord or arcs come first.

Definition 1. ***$\overline{st}$-Composed Convex Hull (st-CCH).*** *Define the minimal convex set that contains the segment $\overline{st}$ and is not intersected by any circle, constructed iteratively by adding circles touched by the current hull and taking their convex hull (see Figure 2).*

Lemma 4. st-CCH containment An optimal path lies within the st-CCH; outside excursions can be shortened by replacing them with hull edges or arcs.

Therefore, we restrict visibility graph construction and sampling to the st-CCH; circles outside the st-CCH can be ignored, as the optimal path will not pass through them.

5.1 Analytic baseline: one circular budget zone

We begin with the case of a single circular budget zone ($n = 1$) and show that an optimal B-legal path can be computed in polynomial time. In our binary per-zone model, if the path enters c_1, then it must consume exactly b_1 units inside it; for clarity, we denote this fixed in-zone chord length by B in the one-zone derivation (so here $B = b_1$). We note that for the case $n = 1$, the constraint $\ell(P \cap c_1) = b_1$ is equivalent to the relaxation $\ell(P \cap c_1) \leq b_1$. This holds because any path P utilizing a budget $b'_1 < b_1$ can be strictly improved (shortened) by increasing the chord length within c_1 until the budget b_1 is exhausted. We assume throughout that $b_1 \leq \ell(\overline{st} \cap c_1)$; otherwise, the direct segment $\overline{st}$ is the trivial solution. If this condition does not hold (i.e., $\ell(\overline{st} \cap c_1) < b_1$), then the optimal solution is to take the straight segment $\overline{st}$.

Let c_1 be a circle of radius r centered at o. Any optimal B-legal path enters c_1 at some point $p_{\text{in}} \in \partial c_1$ (the circumference of c_1), traverses a chord of length B inside c_1, and exits at a point $p_{\text{out}} \in \partial c_1$. The portion of the path outside c_1 consists of straight-line segments from s to p_{in} and from p_{out} to t.

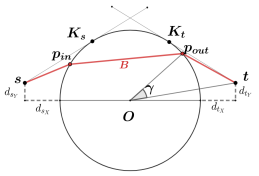

Fig. 3

Let $\gamma := \angle top_{\text{out}}$ (see Fig. 3). By the law of cosines in $\triangle osp_{\text{in}}$ and $\triangle otp_{\text{out}}$, and since the in-circle segment is a fixed chord of length B, we obtained the total length of a feasible path P as a function of γ:

$$\ell(P) = f(\gamma) = \sqrt{r^2 + |os|^2 - 2r|os|\cos(\angle sop_{\text{in}})} + B + \sqrt{r^2 + |ot|^2 - 2r|ot|\cos(\gamma)}.$$

The total path length $\ell(P)$ is a continuous, differentiable function of γ; the optimal path P^* can be efficiently obtained by solving $f'(\gamma) = 0$ over the closed, bounded feasible domains.

5.2 Two circles

Based on the results for a single circular budget zone (Section 5.1), we conclude that for two circular budget zones finding the optimal path requires minimizing a function of two variables, which is significantly more challenging than the single-variable case and remains difficult in more general settings (larger n and varying circle radii). Combined with the proven hardness of the C-BCSP, this motivates the development of approximate algorithms.

6 THE IRD ALGORITHM

We now introduce our Iterative Refinement Discretization (IRD) algorithm and its approximation guarantees. The IRD algorithm is an *anytime* algorithm that begins with a coarse discretization and, at each iteration, refines it only within a narrow corridor around the current best path. This yields progressively improving solutions that converge to the optimal as refinement continues.

We iteratively discretize each circle c_i into k points on ∂c_i with uniform angular spacing $\Delta\theta = 2\pi/k$ (equivalently, arc-length spacing $d = 2\pi r_i/k$). Based on those points, we then construct a visibility graph over s, t, and the discretized boundary points, and solve a resource-constrained shortest path (RCSP) problem to obtain a B-legal solution.

Although the resulting discretized problem is computationally challenging, as RCSP is NP-hard in general, many existing RCSP algorithms perform well in practice on graphs of moderate size. Importantly, their running time scales with the number of vertices and edges in the visibility graph- smaller graphs lead to faster RCSP solutions.

This creates a trade-off: on the one hand, increasing the number of discretization points k improves solution quality, as the approximation error vanishes when $k \to \infty$ and the discretized path length converges to the continuous optimum; on the other hand, the number of vertices grows as $\Theta(nk)$, and the computational complexity increases accordingly. Consequently, a fine uniform discretization may induce unnecessarily large graphs and become the dominant computational bottleneck. This directly motivates our IRD approach, which reduces graph size by iteratively refining the discretization only where needed, achieving efficiency without sacrificing solution quality.

6.1 Bounded deviation in discretized paths

The key insight enabling efficient iterative refinement is that a coarse discretization with small k not only provides an approximate solution, but also *identifies a narrow corridor* containing the optimal continuous path. This allows us to refine the graph only within this corridor, dramatically reducing computational cost while maintaining solution quality.

Theorem 2 (Bounded deviation for n circles). *Let C be n disjoint circles of radius r, each discretized into k equally spaced points with step size $d =$*

$2\pi r/k$. *Let OPT*(d) *denote the optimal path in the discretized model,* P^* *the global continuous optimum (the unconstrained shortest B-legal path from* s *to* t*),* $S \subseteq C$ *the set of circles entered by OPT*(d)*, and* P_S^* *the continuous optimal path* restricted to enter exactly the circles in S. *For each circle* $c_j \in S$*, let* p_{in}^j, p_{out}^j *be the entry and exit points of OPT*(d)*, and* q_{in}^j, q_{out}^j *the corresponding points of* P_S^*. *Then:*

$$\widehat{\ell}\,(p_{in}^j, q_{in}^j) \leq d/2 \quad \textit{and} \quad \widehat{\ell}\,(p_{out}^j, q_{out}^j) \leq d/2,$$

where $\widehat{\ell}\,(\cdot,\cdot)$ *denotes arc length along* ∂c_j. *In particular, when* S *equals the circle set entered by the global continuous optimum* P^**,* $P_S^* = P^*$ *and the bound holds against* P^* *itself.*

Proof. We prove by induction on n (the number of circles). For the base case ($n = 1$), as shown in Section 5.1, the path length function $\ell(P)$ is a continuous, differentiable single-variable function. The optimal continuous entry point corresponds to a minimum of this function (if multiple minima exist with the same value, any one suffices). The discretization point chosen in $\mathrm{OPT}(d)$ must be the one closest to this optimal continuous entry/exit point; otherwise, a shorter path could be found, contradicting optimality. Since discretization points are spaced at distance d along the circle boundary, the closest discretization point lies within distance $d/2$ of the optimal continuous point, yielding the $d/2$ bound.

Inductive assumption: Assume the claim holds for $n-1$ circles. For any instance with $n-1$ circles and discretization step size d, the optimal discretized path from s to any point $\widetilde{t}$ on the boundary of the $(n-1)^{\text{th}}$ circle remains within distance $d/2$ of the corresponding optimal continuous path from s to $\widetilde{t}$.

Inductive step: Consider an instance with n circles, and let c_n be the last circle traversed by the (continuous/discretized) optimal path. Let $\widetilde{t}$ denote an arbitrary candidate exit point on the boundary of c_{n-1}. From $\widetilde{t}$ to the final target t, the remaining subproblem involves at most a single-circle instance (namely, traversal of c_n), which is exactly the base case analyzed for one circle. At first glance, this decomposition is not obviously valid because the objective is not separable across circles due to the global budget coupling.

To address this, we account for all possible allocations of the global budget between the first $n-1$ circles and the last circle. For each candidate point $\widetilde{t}$, we consider both cases: (i) compute an optimal path from s to $\widetilde{t}$ over the first $n-1$ circles with the full global budget B, and concatenate it with the optimal single-circle path from $\widetilde{t}$ to t (which may skip c_n); (ii) compute an optimal path from s to $\widetilde{t}$ with reduced global budget $B - b_n$, corresponding to the case in which the model commits to consuming the zone budget of c_n (i.e., $\ell(P \cap c_n) = b_n$), and concatenate it with the optimal single-circle path from $\widetilde{t}$ to t that traverses c_n.

By the inductive assumption, the discretized entry/exit points for the first $n-1$ circles deviated from the continuous ones by at most $d/2$ for any fixed $\widetilde{t}$, and by applying the one-circle base case to the final subproblem we obtained the same bound for c_n, completing the induction.

This theorem has significant algorithmic implications: the continuous optimal path P^* lies within an arc of length at most d centered at a discretization point

$p \in \mathrm{OPT}(d)$. For a finer discretization $d' = \alpha d$ with $\alpha \in [1/2, 1)$, the optimal path $\mathrm{OPT}(d')$ must pass through one of the three points $p - d', p, p + d'$, each within arc distance d' of p on the circle. This induces a narrow *corridor* around $\mathrm{OPT}(d)$ that contains all finer-resolution optimal paths and enables substantial pruning of the discretization.

Motivated by this property, we propose the Iterative Refinement Discretization (IRD) algorithm. At iteration i, the resolution doubles, $k_i = k_{\text{init}} \cdot 2^i$, but refinement is restricted to this corridor: for each circle, IRD considers only three candidate entry and exit points- the previous iteration's optimal points and their two immediate neighbors. Consequently, the search space per iteration is reduced from $O(k_i^{2n})$ under full discretization to $O(3^n)$ combinations, while retaining solution quality comparable to full discretization.

6.2 A geometric overhead bound

For a circle of radius r discretized with spacing $\Delta\theta$ (arc-length $d = 2\pi r/k$), following theorem 2, the worst-case boundary mismatch between a continuous boundary point and the nearest discrete sample is at most $\Delta\theta/2$ in angle. To bound the discretization error $h(d, n)$, we analyzed the worst-case deviation over a pair of adjacent circles.

Consider two equal-radius circles tangent at a single point (if the radii differ, we consider the larger one), each defining a trapezoid with base along the common diameter (See Figure 4). Let α denote the angle defining the inner trapezoid, and $\theta_d/2$ the discretization angular resolution. The outer trapezoid has angle $\alpha + \theta_d/2$. The maximal deviation in crossing length is $x' - x$, where the upper base lengths are:

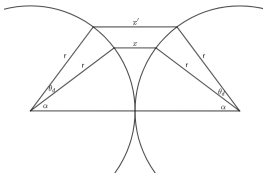

Fig. 4

$$x = 2r(1 - \cos(\alpha)), \quad x' = 2r(1 - \cos(\alpha + \theta_d/2))$$

Maximizing $x' - x = 2r(\cos(\alpha) - \cos(\alpha + \theta_d/2))$ with respect to α yields $x' - x = 4r \sin(\theta_d/4) \approx r\theta_d$ for small θ_d. Aggregating across n circles gives:
$h(d, n) \leq (n + 1) \cdot 4r \sin(\theta_d/4) \approx (n + 1) \cdot r\theta_d$

6.3 The Iterative Refinement Discretization (IRD) Algorithm

Algorithm 1 presents the Iterative Refinement Discretization (IRD) algorithm in detail. The algorithm begins by computing the $\overline{st}$-CCH to prune circles that cannot participate in optimal solutions (Lemma 5). It then constructs a visibility graph over a coarse discretization of the remaining circle boundaries: for each boundary sample, chords of length exactly b_i (the per-zone budget) are added to represent valid in-zone traversals, along with boundary arcs and free-space tangent and segment edges. A resource-constrained shortest path (RCSP [16]) instance is then solved to obtain an initial solution $\mathrm{OPT}(d)$.

In each subsequent iteration, IRD refines the discretization by a factor $\alpha \in [1/2, 1)$, and restricts refinement to the narrow corridor around the current solution: for each traversed circle, only three candidate points (the previous entry/exit point and its two immediate neighbors at the finer resolution) are considered. The algorithm terminates when the maximum arc deviation falls below a user-specified tolerance ϵ, returning the refined optimal path.

As established by Theorem 2, the continuous optimal path P^* lies within arc-length $d/2$ of any discretized solution $\mathrm{OPT}(d)$. This provides IRD with an *anytime* property: at any iteration, the current solution has a known quality bound relative to the true optimum, and continued refinement provably converges to the optimal continuous solution. Since each candidate set is guaranteed to include either the continuous optimal point or its closest discretization neighbor, no optimality is lost during refinement. Moreover, the geometric decay of the deviation ensures convergence from any initial discretization, while finer initial resolutions require fewer refinement iterations to reach a path that is within the stated bound of the optimal one. The search space per iteration remains $O(3^n)$ combinations, compared to the $O(k_i^{2n})$ combinations required by full discretization at resolution k_i, yielding an exponential reduction in computational effort. The bound transfers to the unconstrained continuous optimum P^* once k_{init} is large enough for $\mathrm{OPT}(d)$ to recover the optimal circle set S^* (Theorem 2); empirically, $k_{\text{init}} = 6$ sufficed across all our instances.

The approach extends naturally to varying radii by adapting the discretization step to each circle individually.

Algorithm 1: Iterative Refinement Discretization (IRD) with $\overline{st}$-CCH pruning

Input: Circles C with radii $\{r_i\}$ and per-zone budgets $\{b_i\}$, global budget B, endpoints s, t, initial k, refinement factor $\alpha \in [\frac{1}{2}, 1)$, tolerance ϵ
Output: Approximate optimal path OPT
$C \leftarrow \{c_i \in C : c_i \subseteq \overline{st}\text{-CCH}\}$; `// prune`
$d \leftarrow 2\pi r/k$; build VG-RCSP graph $G = (V, E)$ with k boundary samples per circle, chord edges of length b_i, boundary arcs, and free-space tangents/segments;
$\mathrm{OPT}(d) \leftarrow$ solve RCSP $\min \sum w(e)$ s.t. $\sum g(e) \leq B$ on G;
repeat
  $d' \leftarrow \alpha d$;
  foreach *circle* c_j *traversed by* $\mathrm{OPT}(d)$ **do**
    restrict V_j to the 3 entry candidates $p^j_{\text{in}} + \{-d', 0, d'\}$ and 3 exit candidates $p^j_{\text{out}} + \{-d', 0, d'\}$; rebuild the arc, chord, and tangent edges incident to c_j;
  end
  re-solve RCSP on updated G to get $\mathrm{OPT}(d')$; $d \leftarrow d'$;
until maximum arc deviation $< \epsilon$;
return $\mathrm{OPT}(d)$;

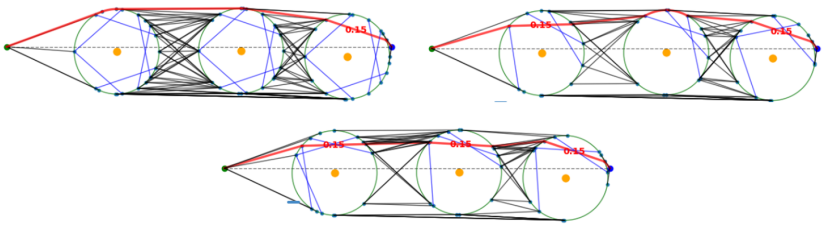

Fig. 5: IRD refinement: the algorithm iteratively refines discretization around the current best path (red), focusing on the narrow corridor containing the optimal solution. Each iteration shows the refinement progress.

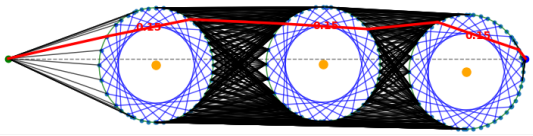

Fig. 6: Full discretization (VG-RCSP at k=24): visibility graph with all boundary points, arcs, chords, and tangents.

7 EMPIRICAL EVALUATION

We empirically compare between three algorithmic approaches for C-BCSP instances with circular budget zones: (1) our proposed iterative refinement discretization method (IRD), initialized with $k_{\text{init}} = 6$, which was selected to balance the number of refinement iterations (though other initial resolutions are also valid); (2) a fully discretized visibility graph with RCSP (VG-RCSP) at dense resolution (specifically, $k = 24$); and (3) a constrained RRT* variant with circle-aware expansion (CC-RRT*) that we developed for this work to test spaces with a large number of circles, as described in detail in Section 7.1. We have fully implemented all algorithm variations, where the code is open and available in `https://github.com/michallazar8/BCSP`.

In our implementation, VG-RCSP builds the full visibility graph with a fixed boundary resolution and solves the resulting resource-constrained shortest path instance using the BiDirectional algorithm from `cspy` [16]; IRD uses the same graph primitives but starts from a coarse discretization and iteratively refines only near the entry/exit samples used by the current best solution. For varying circle radii, IRD terminates when each circle achieves a target perimeter spacing based on the smallest radius: we set $\ell_{\text{arc}} = 2\pi r_{\min}/k_{\text{goal}}$ with $r_{\min} = \min_i r_i$, and refine circle c_i until its boundary spacing is at most ℓ_{arc}.

Our experimental evaluation proceeds in two phases: first, we compare all three methods on instances with uniform-radius circles to establish baseline per-

formance and validate that IRD achieves solution quality comparable to full discretization while reducing computational cost. Second, we compare IRD and CC-RRT* on instances with varying-radius circles to test scalability in more realistic scenarios.

7.1 Constrained RRT* with circle-aware expansion (CC-RRT*)

The constrained RRT* variant extends standard RRT* [9] to handle budget constraints and circular zone boundaries. The algorithm maintains a tree rooted at s and iteratively expands it toward the goal t through random sampling (see illustration in Fig. 7).

Tree expansion. At each iteration, the algorithm samples x_{rand}, finds the nearest tree node x_{near}, and generates a candidate point x_{new} in the direction of x_{rand} with step size η. If x_{new} lies inside some circle c_i, we do not insert it; instead we create a boundary node at the first intersection of $\overline{x_{\text{near}} x_{\text{new}}}$ with ∂c_i (unless one already exists), and spawn two symmetric perimeter nodes offset by a chord distance corresponding to the per-zone budget b_i.

Parent selection and rewiring. We select the best parent for x_{new} among nodes within radius $\gamma_n = \gamma(\log n/n)^{1/2}$, and then apply standard RRT* rewiring, accepting connections only if they preserve budget feasibility.

Budget feasibility. Each path from s to any node must satisfy the global constraint $\sum_i \ell(P \cap c_i) \leq B$ and the per-zone constraints $\ell(P \cap c_i) \in \{0, b_i\}$. We enforce feasibility checks when adding edges (and before connecting to the goal).

Termination. CC-RRT* terminates once each circle accumulates $k_i/2$ entry points (yields a total of $(3/2)\, k_{\text{goal}}$ boundary nodes), where $k_i = \lceil 2\pi r_i/\ell_{\text{arc}} \rceil$ and $\ell_{\text{arc}} = 2\pi r_{\min}/(k_{\text{goal}}/1.5)$.

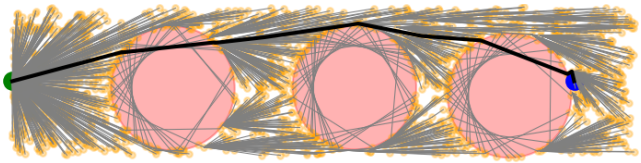

Fig. 7: CC-RRT* visualization: the algorithm constructs a tree (orange nodes and grey edges) through random sampling, with the optimal path highlighted in black. The algorithm adapts to circular budget zones by creating boundary nodes and maintaining budget feasibility constraints.

7.2 Experimental Protocol

To ensure fair comparison, all three methods were evaluated on identical problem instances with the same: circle configurations (positions, radii), per-zone budget distribution $\{b_1, \ldots, b_n\}$, global budget B, target discretization resolution $k_{\text{goal}} = 24$, and start/target points s and t. Each experiment was run with

30 different random seeds for each number of circles n, and the reported results correspond to the mean over these runs.

Performance metrics. We evaluate each algorithm using: **(1) solution quality** - total path length $\ell(P)$; **(2) feasibility** - per-zone budgets in $\{0, b_i\}$ and global budget B; **(3) computational efficiency** - runtime and resulting graph/tree size (nodes and edges); and **(4) scalability** - performance trends as n and the circle radii vary.

7.3 Uniform radius experiments

We first compare all three methods on instances with circles of uniform radius. This setup allows us to assess baseline performance and validate that IRD achieves solution quality comparable to full discretization while reducing computational cost.

Results Figure 8 further confirms that the three methods produce very similar mean path lengths across n. Figure 9 shows that IRD with refinement from k_{init}=6 to k_{goal}=24 achieves the near-baseline solution quality while using much smaller graphs, and remains practical at larger n values, where the fully discretized VG-RCSP quickly becomes prohibitively expensive. The fully discretized baseline grows rapidly in both edges and runtime (reaching thousands of seconds already around n=6–7). CC-RRT* scales more smoothly in runtime than full discretization, but requires orders of magnitude more nodes.

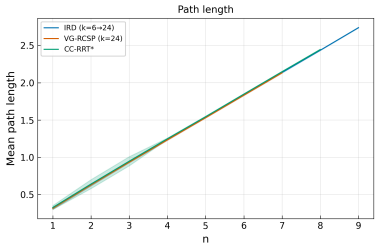

Fig. 8: Uniform-radius experiments: mean path length as a function of n.

7.4 Varying radius experiments

We compare IRD and CC-RRT* on instances with circles of varying radii. This configuration tests scalability to scenarios where circle sizes differ, and evaluates how each method adapts to non-uniform geometry. Note that in our CC-RRT* implementation for these experiments, due to implementation considerations, budget feasibility constraints are checked when connecting to the goal t, rather than at every edge addition during tree expansion.

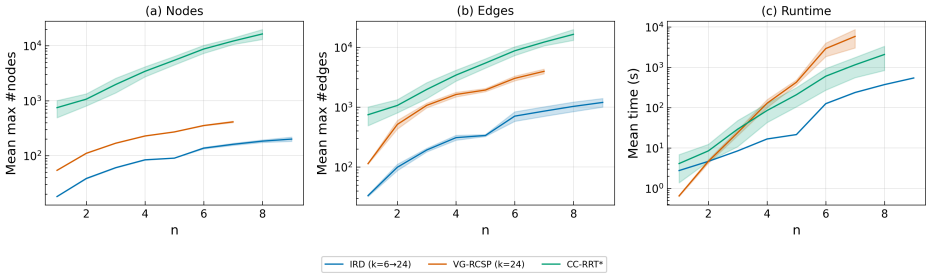

Fig. 9: Uniform-radius experiments: mean maximum graph/tree size (nodes, edges) and mean runtime as a function of n. Log scale on y-axis.

Results Figure 11 demonstrates that IRD maintains its efficiency advantage over CC-RRT* even when circle radii vary, and reveals that CC-RRT*'s runtime grows dramatically worse in non-uniform settings compared to uniform-radius instances. While CC-RRT* requires substantially more nodes and edges (growing to tens of thousands of nodes for $n \geq 4$), IRD keeps graph sizes modest and runtime manageable. The runtime gap widens significantly as n increases, with CC-RRT* exceeding 8700 seconds for $n = 6$ compared to IRD's approximately 173 seconds. We note that we tested different step-sizes, the relative performance trends remain consistent across these variations. Figure 10 shows that IRD achieves slightly better (shorter) path lengths than CC-RRT*, confirming that IRD's efficiency gains do not come at the cost of solution quality in non-uniform settings - indeed, IRD delivers both superior efficiency and superior solution quality.

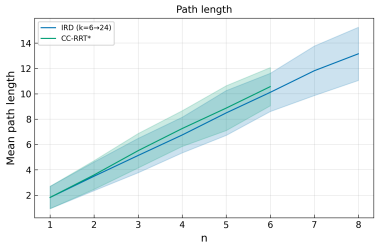

Fig. 10: Varying-radius experiments: mean path length as a function of n.

Effect of ST-CCH sampling on CC-RRT.* Algorithm CC-RRT*, as described in Section 7.1 and implemented in the experiments, samples x_{rand} from the bounding rectangle of $\{s, t\} \cup C$. Since Lemma 5 states that an optimal path is necessarily contained in the ST-CCH, we also tested a variant that samples only from the ST-CCH, enlarged by a margin $\delta \geq 0$. This enlargement allows the tree to route around circles whose boundaries touch the ST-CCH. On a fixed varying-radius instance with $n = 5$, the ST-CCH variant at $\delta = 0.3$ yielded a

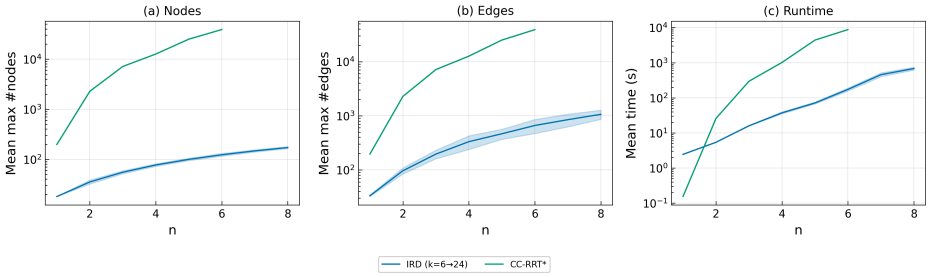

Fig. 11: Varying-radius experiments: mean maximum graph/tree size (nodes, edges) and mean runtime as a function of n. Log scale on y-axis.

$\sim 3.3\times$ speed-up, $\sim 40\%$ fewer tree nodes, and four successive path refinements vs. only one for the rectangle baseline at comparable final length. However, for $\delta = 0.1$ the gain essentially vanished, demonstrating that δ must be carefully tuned to the instance.

This comparison is subject to two important caveats: First, B-legality is enforced only at goal-commit (Section 7.1); checks at every CC-RRT* extension and rewire makes the runtime prohibitive within our current implementation framework. Second, the best ST-CCH inflation δ for CC-RRT* is strongly instance-dependent.

8 3D EXTENSION

We show in this section that the bounded-deviation theorem and the narrow-corridor property, stated and proven for planar circular budget zones, extend directly to spherical budget zones in $\mathbb{R}^3$.

Dependence on the base case. Examination of the bounded-deviation proof reveals that the induction over budget zones relies on a single geometric property of the base case (n=1): the total path length is a continuous function of the boundary entry location, so the discretized optimum must select a boundary point within arc-length distance at most $d/2$ from the continuous one.

Spherical base case. Consider a single spherical budget zone $B = \{X \in \mathbb{R}^3 : \|X - o\| \le r\}$ with source/target $s, t \notin B$ and in-sphere budget $b \in (0, 2r)$. Any feasible path $\Pi = s \to P \to Q \to t$ with $P, Q \in \partial B$ satisfies the chord constraint $\|P - Q\| = b$. Parameterizing $P \in \partial B$ by spherical angles $(\alpha, \beta) \in [0, 2\pi] \times [0, \pi]$ via $P(\alpha, \beta) = o + r\, u(\alpha, \beta)$ with $u(\alpha, \beta) = (\cos\alpha \sin\beta, \sin\alpha \sin\beta, \cos\beta)^\top$, the chord subtends a fixed central angle $\theta = 2 \arcsin(b/2r)$, lies in the plane through o spanned by u and $\overrightarrow{ot}$, and the exit point $Q(\alpha, \beta)$ is obtained by rotating u by θ about the normal to this plane (Rodrigues' formula). The base-case objective is therefore

$$f(\alpha, \beta) \;=\; \|s - P(\alpha, \beta)\| \;+\; b \;+\; \|Q(\alpha, \beta) - t\|,$$

a sum of norms and a constant, continuous on the compact domain $[0, 2\pi] \times [0, \pi]$ and hence attaining a global minimum.

Implications. Discretizing ∂B with step d, continuity of f implies the discretized minimum differs from the continuous one by at most $O(d)$, so the bounded-deviation induction extends verbatim to spherical zones and IRD applies unchanged in $\mathbb{R}^3$. **More generally, the bounded-deviation theorem and IRD extend to any domain geometry for which the single-zone base-case objective is a continuous function of the boundary parameterization.** The full derivation with explicit geometric constructions is provided in the supplementary material.

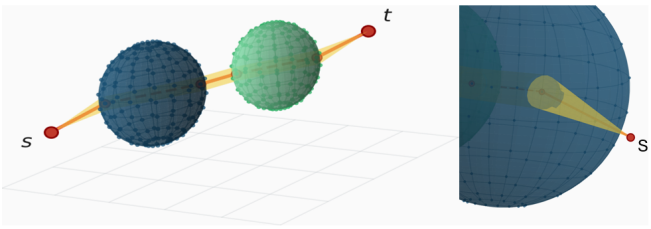

Fig. 12: Illustration of the spherical budget zone extension in $\mathbb{R}^3$.

9 CONCLUSION AND FUTURE WORK

In this paper, we formalized the Circular Budget-Constrained Shortest Path (C-BCSP) problem, proved its computational hardness, and introduced IRD - an iterative refinement algorithm exploiting a bounded deviation theorem to reduce the search to only three candidate entry/exit points per circle while provably converging to the optimal solution. Experiments show IRD matches the solution quality of full discretization and an RRT*-based algorithm (CC-RRT*) with over an order-of-magnitude runtime improvement and substantially smaller graph sizes. Finally, we have shown that our results extend to spherical zones in $\mathbb{R}^3$.

The work opens many core questions for future research, including extensions to more general convex geometries and non-circular budget zones, where boundary structure and entry–exit behavior may differ. Additional directions include multi-robot coordination under shared or competing budget constraints, where robots may be required to maintain formations or satisfy inter-robot distance constraints, as well as online planning in dynamic environments, and tighter approximation guarantees under adaptive discretization schemes. Exploring these directions would further broaden the applicability of budget-constrained continuous path planning in realistic robotic settings.

Acknowledgments. This study was funded in part by ISF grant number 1563/22.

References

1. Agarwal, P.K., Kumar, N., Sintos, S., Suri, S.: Computing shortest paths in the plane with removable obstacles. In: 16th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2018) (2018)
2. Bhattacharya, P., Gavrilova, M.L.: Roadmap-based path planning-using the voronoi diagram for a clearance-based shortest path. IEEE Robotics & Automation Magazine **15**(2), 58–66 (2008)
3. Davoodi, M., Enamzadeh, H., Safari, A.: Path planning in a weighted planar subdivision under the manhattan metric. In: CCCG. pp. 80–86 (2020)
4. Devaurs, D., Siméon, T., Cortés, J.: Enhancing the transition-based RRT to deal with complex cost spaces. In: 2013 IEEE International Conference on Robotics and Automation (ICRA). pp. 4120–4125. IEEE (2013)
5. Dumitrescu, I., Boland, N.: Algorithms for the weight constrained shortest path problem. International Transactions in Operational Research **8**(1), 15–29 (2001)
6. Erickson, L., LaValle, S.: A simple, but np-hard, motion planning problem. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 1388–1393 (2013)
7. Gammell, J.D., Srinivasa, S.S., Barfoot, T.D.: Batch informed trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 3067–3074. IEEE (2015)
8. Gang, L., Wang, J.: Prm path planning optimization algorithm research. Wseas Transactions on Systems and control **11**(81-86), 7 (2016)
9. Karaman, S., Frazzoli, E.: Incremental sampling-based algorithms for optimal motion planning. In: Robotics: Science and Systems VI (2011)
10. Marcucci, T., Umenberger, J., Parrilo, P., Tedrake, R.: Shortest paths in graphs of convex sets. SIAM Journal on Optimization **34**(1), 507–532 (2024)
11. Mitchell, J., Papadimitriou, C.: The weighted region problem. In: Proceedings of the third annual symposium on computational geometry. pp. 30–38 (1987)
12. Öztürk, D.T., Köksalan, M.: Biobjective route planning of an unmanned air vehicle in continuous space. Transportation Research Part B: Methodological **168**, 151–169 (2023)
13. Pachter, M., Hebert, J.: Minimizing radar exposure in air vehicle path planning. IFAC Proceedings Volumes **35**(1), 181–185 (2002)
14. Pfeiffer, B., Batta, R., Klamroth, K., Nagi, R.: Path planning for uavs in the presence of threat zones using probabilistic modeling. IEEE Trans. Autom. Control **43**, 278–283 (2005)
15. Piatko, C.D.: Geometric bicriteria optimal path problems. Cornell University (1993)
16. Sanchez, D.T.: cspy: A python package with a collection of algorithms for the (resource) constrained shortest path problem. Journal of Open Source Software **5**(49), 1655 (2020). https://doi.org/10.21105/joss.01655, `https://doi.org/10.21105/joss.01655`
17. Wilson, T.S., Thomason, W., Kingston, Z., Gammell, J.D.: AOR-RTC: Almost-surely asymptotically optimal planning with RRT-connect. IEEE Robotics and Automation Letters **10**(12), 13375–13382 (2025). https://doi.org/10.1109/LRA.2025.3615522
18. Wolek, A., Cheng, S., Dey, B., Feron, E., Goel, J., Hovakimyan, N., Raz, E., Savas, Y., Shishika, D.: Sampling-based risk-aware path planning around dynamic engagement zones. arXiv preprint arXiv:2407.03644 (2024)