Algorithm for Optimal Chance Constrained Knapsack with Applications to Multi-robot Teaming

Fan Yang[1] and Nilanjan Chakraborty[2]

***Abstract*— Motivated by applications in multirobot team selection, in this paper, we present a novel algorithm for solving chance-constrained 0-1 knapsack problem, where the objective function is deterministic but the weights of the items are stochastic and therefore the knapsack constraint is stochastic. We convert the chance-constrained knapsack problem to a two-dimensional discrete optimization problem on the variance-mean plane, where each point on the plane can be identified with an assignment of items to the knapsack. By exploiting the geometry of the non-convex feasible region of the chance-constrained knapsack problem in the variance-mean plane, we present a novel deterministic technique to find an optimal solution by solving a sequence of deterministic knapsack problems (called risk-averse knapsack problem). We apply our algorithm to a multirobot team selection problem to cover a given route, where the length of the route is much larger than the length each individual robot can fly and the length that an individual robot can fly is a random variable (with known mean and variance). We present simulation results on randomly generated data to demonstrate that our approach is scalable with both the number of robots and increasing uncertainty of the distance an individual robot can travel.**

I. Introduction

The knapsack problem is a fundamental problem in combinatorial optimization that has multiple applications in task allocation and team formation in multi-robot systems. For example, in algorithms to solve the generalized assignment problem for multiple robots, the knapsack problem is a subproblem that needs to be solved multiple times [6]. In this paper, we consider a multirobot team formation problem, where we consider a group of heterogeneous robots that has to cover a given route with known length. Each robot has a limited battery life and therefore there is a upper limit on the distance that the robot can travel. Furthermore, the travel distances are uncertain because they depend on uncertain environmental variables like wind speed. We assume that the lengths that robots can travel are independent Gaussian random variables with known means and variances. There is operating cost for each robot. The total cost of covering the route is a sum of individual costs of robots. Our goal is to find a team of robots with the minimum total cost that covers the route with high probability (specified *a priori*).

The deterministic version of our problem where the travel distances are known constants can be formulated as a 0-1 knapsack problem. There are many methods to solve the knapsack problem such as dynamic programming [12], branch and bound method [9] and other methods that combine both methods [10], [7], [8]. Although solving knapsack problem is NP-hard, there is a fully polynomial time approximation scheme [12]. There are different stochastic variations of the classical 0-1 knapsack problem that have been studied in the extant literature. In [1], [4], [11], the authors have studied the stochastic knapsack problem with deterministic weights and random costs whereas in our problem we have deterministic costs and random weights. In [2], the authors compute a solution policy that optimize the expected total values. Optimizing expected values provide no performance guarantees on a particular realization of the random variables. We want to develop methods that ensures the constraints are satisfied with a high probability irrespective of the realization of the random weights. An algorithm is designed to obtain good solutions to the chance-constrained problem in [5], by running a sequence of robust problems. The algorithm provides an optimal solution when the costs are identical or the uncertain weights present all the same characteristic. In this paper, our method computes the optimal solution in more general situation. In [3], the authors consider a stochastic knapsack problem similar to our setting and provide a polynomial time approximation scheme (PTAS) by using a parametric linear programming reformulation. Our solution to the chance-constrained problem is based on a geometric interpretation of the problem on variance-mean plane. Our method finds the optimal solution of chance-constrained problem by solving a sequence of a deterministic knapsack problems called the risk-averse knapsack problems.

Contributions: In this paper, we present a novel algorithm that solves 0-1 knapsack problem with chance constraint. By analyzing the feasible region of both chance-constrained and risk-averse knapsack problems on variance-mean plane, we prove that there exists a risk-averse knapsack problem such that the optimal solution of chance constrained knapsack problem is also the optimal solution of risk-averse knapsack problem. We use this insight to develop an iterative algorithm where we solve the chance constrained problem by repeatedly solving a sequence of risk-averse knapsack problem. The key aspect of our algorithm is that we maintain a probabilistic guarantee irrespective of the realization of the random variables (the lengths the robots could move). We present simulation results on randomly generated data which show that our algorithm works efficiently. An extended version of this paper with the proofs and more elaborate simulation results is under review at *IEEE International Conference on Robotics and Automation*, 2018.

[1]Fan Yang and [2]Nilanjan Chakraborty are with the Mechanical Engineering Department at SUNY, Stony Brook. Email: `fan.yang.3@stonybrook.edu`, `nilanjan.chakraborty@stonybrook.edu`

II. Chance Constrained Knapsack Problem

Let L be the length of the closed curve (or a route) that a team of robots have to cover. We have a collection of heterogeneous robots that have different battery life and they can fly for different lengths. Let ℓ_i be the distance that robot i can fly. Each robot has a different operating and maintenance cost denoted by c_i. The variable ℓ_i is assumed to be a Gaussian random variable with mean μ_i and variance σ_i^2, i.e., $\ell_i \sim \mathcal{N}(\mu_i, \sigma_i^2), i = 1, ..., n$. Our goal is to find a set of robots from the collection of n robots that can cover the total length L with probability p (where $0 \leq p \leq 1$) while minimizing the total cost. Let f_i be an integer variable that takes the value 1 if robot i is part of the team and 0 otherwise. The integer program formulation of our problem is:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} c_i f_i \\ \text{s.t.} \quad & \mathbb{P}\left(\sum_{i=1}^{n} \ell_i f_i \geq L\right) \geq p \\ & f_i \in \{0,1\}, \quad \forall i = 1, \ldots, n \end{aligned} \tag{1}$$

If we relax f_i, the problem in (1) is a second order cone program with integrality gap $\Omega(\sqrt{n})$ [3].

Lemma 1: The CC-KAP problem in (1) with a given probability p is equivalent to the following formulation

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} c_i f_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} \mu_i f_i - C\sqrt{\sum_{i=1}^{n} \sigma_i^2 f_i} \geq L \\ & f_i \in \{0,1\}, \quad \forall i = 1, \ldots, n \end{aligned} \tag{2}$$

where $C = \Phi^{-1}(p)$ is a constant.

In [3], the authors converted the problem in Equation (2) to a parametric linear program and presented an algorithm that for $\epsilon > 0$ gives a $1 - 3\epsilon$ approximate solution with running time $O\left(\frac{1}{\epsilon^2} n^{\frac{1}{\epsilon}}\right)$. We present an alternate parametric formulation, where different choices of the parameter leads to different knapsack problems. In the discussion below we will refer to both (1) and (2) as chance constrained knapsack problem (CC-KAP), which is a chance constrained integer optimization problem and is hard to solve in general. In this paper, instead of solving CC-KAP directly, we show that the solution to CC-KAP can be obtained by solving a number of deterministic knapsack problems (given below), which we call risk-averse knapsack problem (RA-KAP)

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} c_i f_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} \mu_i f_i - \lambda \sum_{i=1}^{n} \sigma_i^2 f_i \geq L' \\ & f_i \in \{0,1\}, \quad \forall i = 1, \ldots, n \end{aligned} \tag{3}$$

Here λ is the risk-averse parameter that performs a weighted combination of the mean and variance of the travel lengths of each robot. The parameter L' is the constraint for the total length in RA-KAP.

III. Geometric Interpretation

In this section, we present a geometric interpretation of the CC-KAP on the variance-mean plane in which the horizontal axis is the variance and the vertical axis is the mean (see Figure 1). The CC-KAP is an integer optimization problems in which any solution is a vector of binary decision variables f_i. Given any particular solution $s = [f_1, ..., f_n]$, we can identify this solution with a point on the variance-mean plane. The y-coordinate of this point is the sum of means for all travel distance of robots chosen in the solution, $\sum_i^n \mu_i f_i$, and the x-coordinate is the sum of variances, $\sum_i^n \sigma_i^2 f_i$. The coordinate of this point related to solution s is denoted by $(\sigma^2(s), \ \mu(s))$. Thus the space of all possible robot teams can be identified with points in the variance-mean plane (however, we do not construct this explicitly because the number of such points will be exponential in the number of robots). Moreover, we can find the feasible region of solution for the CC-KAP based on the chance constraint in Formulation 2. As it is shown in Figure 1, the feasible region for CC-KAP, denoted by $\mathcal{C}$, is the space above the parabola in the first quadrant on variance-mean plane. Since the constraint in the RA-KAP is a linear inequality of σ^2 and μ, the feasible region for RA-KAP, denoted by $\mathcal{R}$, is the space above the line whose slope is equal to risk-averse parameter λ and y-intercept is equal to the length of the route L'. Based on this geometric viewpoint, we present the following lemmas (without proof).

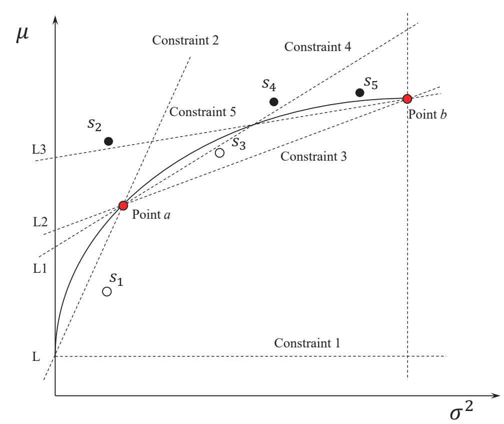

Fig. 1. The geometric interpretation of chance-constrained knapsack problem. Any solution is related to a point on variance-mean plane. The feasible region for CC-KAP is the space above the parabola while the feasible region for RA-KAP with given λ and L is the space above the line whose slope is equal to λ and y-intercept is equal to L.

Lemma 2: The optimal solution of RA-KAP that satisfies the chance constraint provides an upper bound of optimal solution of CC-KAP.

Lemma 3: There exists a RA-KAP, with some choice of λ and L' such that the optimal solution of CC-KAP is also the optimal solution of the RA-KAP.

IV. Algorithm description

Let the intersection of the feasible region of CC-KAP and the risk-averse problem, RA-KAP, in the first quadrant be $\mathcal{I}$ ($\mathcal{I} = \mathcal{C} \cap \mathcal{R}$). Lemma 2 implies that the optimal solution of RA-KAP that satisfies the chance constraints is also optimal for CC-KAP over the feasible region of CC-KAP restricted to $\mathcal{I}$. In this paper we say that the region $\mathcal{I}$ is "explored". To explore the whole feasible region of CC-KAP, namely $\mathcal{C}$, we solve multiple RA-KAPs, whose optimal solution satisfy the chance constraint and corresponding feasible regions ($\mathcal{R}_j$) cover the feasible region of chance-constrained problem, i.e., $\mathcal{C} \subset \bigcup_{j \in S} \mathcal{R}_j$ where S is the index set of RA-KAPs whose optimal solutions satisfy the chance constraint.

The first step starts with solving RA-KAP with $\lambda = 0$ and $L' = L$. If the optimal solution satisfies the chance constraint, as shown in Figure 1 s_1 is above the parabola. The algorithm terminates because $\mathcal{C} \subset \mathcal{R}$. Otherwise, the algorithm computes the constraint of the RA-KAP for the next iteration by equation $\lambda' = C/\sigma$ where $\sigma = \sqrt{\sum_{i=1}^{n} \sigma_i^2 f_i}$. On the variance-mean plane, we can treat the updating procedure as the constraint line rotating clockwise about y-intercept $(0,\ L)$. The new constraint line will be guaranteed to be located above the previous point. The procedure continues until we obtain a feasible solution of chance-constrained problem or the RA-KAP is not feasible, i.e., $\sum_{i=1}^{n} \mu_i - \lambda \sum_{i=1}^{n} \sigma_i^2 < L$. Now we can conclude that the subset of feasible region $\mathcal{C}$, denoted by $\mathcal{I}_I$ is explored although the risk-averse problem might not be a feasible problem since there is no solution in $\mathcal{R}_I$ and $\mathcal{I}_I \subset \mathcal{R}_I$. Note that all solutions are located to the left of the vertical line $\sigma^2 = \sum_{i=1}^{n} \sigma_i^2$ (the line going through Point b in Figure 1) because $\sum_{i=1}^{n} \sigma_i^2 f_i \leq \sum_{i=1}^{n} \sigma_i^2$ where $f_i \in \{0, 1\}\ \forall i$. Let the subset of feasible region of CC-KAP that is on the right hand side of vertical line be $\mathcal{C}_l$. The remaining feasible region is $\mathcal{C} \backslash (\mathcal{I}_I \cup \mathcal{C}_l)$ denoted by $\mathcal{C}'$.

In the second step, the algorithm computes the intersection of parabola and the constraint line of the last RA-KAP in the first step (Point a in Figure 1) and intersection of parabola and the vertical line (Point b in Figure 1). Then we solve the RA-KAP with new constraint obtained by connecting those two points. Since $\mathcal{C}' \subset \mathcal{I}$, the algorithm terminates if the optimal solution of RA-KAP satisfies the chance constraint, e.g., point s_2 or s_4 in Figure 1. Otherwise, we compute two new constraints for two new RA-KAPs. The new constraints should be selected so that the solutions of RA-KAP is different from previous RA-KAP solutions that do not satisfy chance constraints. Moreover, the feasible regions of the new RA-KAPs, say $\mathcal{I}_a$ and $\mathcal{I}_b$, should cover the feasible regions of the current chance-constrained problem, i.e., $\mathcal{I} \subset (\mathcal{I}_a \cup \mathcal{I}_b)$. For example, in Figure 1, s_3, the solution of RA-KAP with constraint 3 is not feasible to CC-KAP. Therefore we compute new constraints 4 and 5. Our procedure guarantees that $\mathcal{I}_3 \subset (\mathcal{I}_4 \cup \mathcal{I}_5)$ and s_3 is not the solution of RA-KAPs with constraint 4 and constraint 5. If the solution of RA-KAP with any generated constraint j does not satisfy the chance constraint, new constraints will be generated based on constraint j. We then solve the RA-KAP with these constraints. If the solution of RA-KAP with a constraint j satisfies the chance constraint, we obtain the optimal solution in $\mathcal{I}_j$ and therefore there is no need to generate new constraints from j. If the RA-KAP with constraint j is not a feasible problem, we do not need to generate new constraints since there is no solution in $\mathcal{I}_j$. The second step terminates when there is no new constraint generated. Thus, we explore the whole feasible region of CC-KAP because $\mathcal{C}' \subset \bigcup_{j \in S} \mathcal{R}_j$ and $\mathcal{C} = \mathcal{C}_I \cup \mathcal{C}_l \cup \mathcal{C}'$. The optimal solution is the one with the smallest objective value among all feasible solutions of chance-constrained problem computed by solving risk-averse problems.

We claim that our algorithm stops in finite number of iterations. For the first step, λ increases till the solution of RA-KAP satisfy the chance constraint. In the worst case, λ will keep increasing until it exceeds a finite bound of λ for which the RA-KAP is not feasible. For the second step, the new constraints will be generated if the solution of RA-KAP is not feasible to chance constraint. The number of solutions is finite and the algorithm prevents obtaining the previous solutions that are not feasible to chance constraint. Therefore in the worst case, the algorithm finds all solutions that are not feasible with a finite number of iterations. In the next section, we present empirical evidence that the above algorithm terminates in a constant number of iterations irrespective of the number of robots.

V. Simulation Results

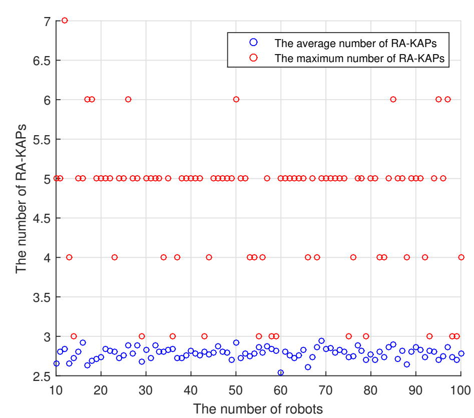

Fig. 2. The total number of RA-KAPs required for solving CC-KAP. The length of the route is 10000 meters and the number of robots vary from 10 to 100. Each data point is obtained from 100 simulations with randomly generated mean and variance of travel distance of each robot

We present simulation studies to understand the scalability of our algorithm as the number of robots increase and as the knowledge about the distance the robots can travel become more uncertain (i.e., the variance increases). To understand the effects of parameters such as the number of robots and the variance of travel distance of robots, we generated different scenarios based on randomly generated parameter values. We first present results for simulations in which the mean and variances of travel distance of robots are randomly generated and the number of robots is varied methodically. Figure 2 and Figure 3 show the performance of our algorithm with the different number of robots and uncertainty in the travel length of robots. The results indicate that the number of robots does not have a significant influence on the speed of our algorithm and the number of calls to RA-KAPs is nearly a constant. In

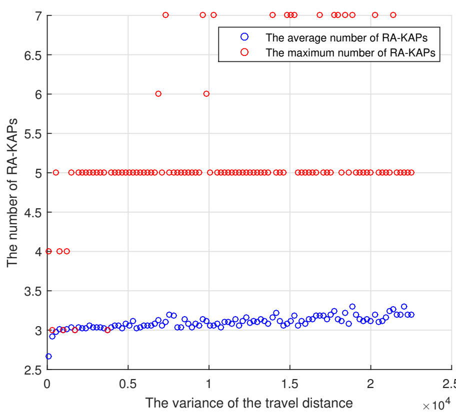

Fig. 3. The total number of RA-KAPs that are used for solving CC-KAP. The number of robots is 100, the length of the route is 50000 meters, and the variance of travel distance vary from 100 to 22500. Each data point is obtained from 100 simulations with randomly generated mean of travel distance of each robot.

the first simulation, we test the effect of number of robots on the number of RA-KAP to be solved, which influences the algorithm performance. The algorithm to solve the RA-KAP is dynamic programming that solves the knapsack problem optimally in pseudo-polynomial time $\mathcal{O}(n^2P)$ where n refers to the number of robots and P refers to the largest cost among all robots [12]. The means and variances of the travel lengths for each robot are generated independently from a uniform distribution $\mu_i \sim \mathcal{U}(1000, 3000)$ and $\sigma_i^2 \sim \mathcal{U}(10000, 12500)$. The length of the curve, L, is 10000 meters and $p = 0.99$. The operation and maintenance cost for each robot is distributed randomly in uniform distribution from 50 to 150, i.e., $c_i \sim \mathcal{U}(50, 150)$. Set $\epsilon = 1 \times 10^{-7}$. We count the number of RA-KAPs for solving CC-KAP when number of robots is equal to $10, 11, \ldots, 100$. For each case with a given number of robots, we generate the means and variances randomly for 100 times.

Figure 2 shows the performance of our algorithm with different number of robots. The results show that the number of calls to RA-KAPs is nearly a constant irrespective of the number of robots. The blue dots represent the average number of RA-KAPs required to solve CC-KAP while the red dots represent the maximum number of calls to RA-KAPs from 100 simulations. The average numbers of RA-KAP solved is almost constant (between 2.5 to 3) irrespective of the number of robots. In Figure 2, the maximum numbers of RA-KAPs solved are between 3 and 7. We observe that maximum number of deterministic knapsack problems solved is 7 which is a small value for application in practice.

In the second simulation, we obtain the effect of the uncertainty of travel distance of robot on the performance of our algorithm by counting the number of calls to RA-KAPs and the actual running time for our algorithm solving CC-KAP with variance equal to $100, 324, 548, \ldots, 22500$. For each case, we generate mean of travel distance of robot randomly based on $\mathcal{U}(1000, 3000)$ for 100 times. The number of robots is 100 and the length of the route is 50000 meters for all scenarios in this simulation. The other parameters such as the cost, probability and ϵ are same as the parameters in the first simulation. Figure 3 shows the average number of calls to RA-KAPs is practically constant as the variance of travel distance increases. The maximum numbers of calls are within the range from 3 to 7.

VI. Conclusion

We presented a novel deterministic algorithm for chance-constrained knapsack problem with application in multi-robot routing with the uncertain travel distances (weights). The key idea in our approach is to convert CC-KAP to a deterministic discrete optimization problem on the variance-mean plane, where each point on the plane can be identified with an assignment of items to the knapsack. By exploiting the geometry of the non-convex feasible region of the CC-KAP in the variance-mean plane, we showed that CC-KAP can be solved optimally by solving a sequence of deterministic knapsack problems (called risk-averse knapsack problem). We demonstrated empirically that our algorithm is quite efficient in practice. Future work includes theoretical complexity bounds of our algorithm.

References

[1] R. L. Carraway, R. L. Schmidt, and L. R. Weatherford. An algorithm for maximizing target achievement in the stochastic knapsack problem with normal returns. *Naval Research Logistics (NRL)*, 40(2):161–173, 1993.

[2] B. C. Dean, M. X. Goemans, and J. Vondrk. Approximating the stochastic knapsack problem: The benefit of adaptivity. *Mathematics of Operations Research*, 33(4):945–964, 2008.

[3] V. Goyal and R. Ravi. A ptas for the chance-constrained knapsack problem with random item sizes. *Operations Research Letters*, 38(3):161 – 164, 2010.

[4] M. I. Henig. Risk criteria in a stochastic knapsack problem. *Operations Research*, 38(5):820–825, 1990.

[5] O. Klopfenstein and D. Nace. A robust approach to the chance-constrained knapsack problem. *Operations Research Letters*, 36(5):628 – 632, 2008.

[6] L. Luo, N. Chakraborty, and K. P. Sycara. Distributed algorithm design for multi-robot generalized task assignment problem. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, November 3-7, 2013*, pages 4765–4771, 2013.

[7] S. Martello, D. Pisinger, and P. Toth. Dynamic programming and strong bounds for the 0-1 knapsack problem. *Manage. Sci.*, 45(3):414–424, Mar. 1999.

[8] S. Martello and P. Toth. A mixture of dynamic programming and branch-and-bound for the subset-sum problem. *Management Science*, 30(6):765–771, 1984.

[9] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Inc., New York, NY, USA, 1990.

[10] V. Poirriez, N. Yanev, and R. Andonov. A hybrid algorithm for the unbounded knapsack problem. *Discrete Optimization*, 6(1):110 – 124, 2009.

[11] M. Sniedovich. Preference order stochastic knapsack problems: Methodological issues. *The Journal of the Operational Research Society*, 31(11):1025–1032, 1980.

[12] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag New York, Inc., New York, NY, USA, 2001.