随机利润背包问题的进化多目标算法

预印本

Kokila Perera
优化与物流
计算机与数学科学学院
阿德莱德大学
阿德莱德，澳大利亚

Aneta Neumann
优化与物流
计算机与数学科学学院
阿德莱德大学
阿德莱德，澳大利亚

Frank Neumann 优化与物流 计算机与数学科学学院 阿德莱德大学 阿德莱德，澳大利亚

arXiv:2303.01695v1 [cs.NE] 3 Mar 2023

摘要

已有大量研究表明，在各种随机组合优化问题中，采用进化多目标算法能够取得良好效果。机会约束优化在复杂的现实场景中发挥着重要作用，因为它允许决策者将环境的不确定性纳入考虑。我们研究了一个具有随机利润的背包问题变体，以保证解的利润达到一定的置信水平。我们介绍了利润机会约束背包问题的多目标形式，并设计了三种双目标适应度评估方法，这些方法独立于所需的具体置信水平。我们使用知名的多目标进化算法 GSEMO 和 NSGA-II 对所提出的方法进行了评估。此外，我们还为 GSEMO 引入了一种过滤方法：该方法根据解的置信水平，定期从中间种群中移除某些解，从而提高最终种群的质量。我们在多个基准测试上展示了所提出方法的有效性，涵盖两种情形：背包物品具有固定的均匀不确定性，以及物品的不确定性与其期望利润呈正相关。

关键词 多目标优化、随机背包问题、机会约束问题

1 引言

现实世界优化问题通常涉及由问题中的某些随机成分或其环境中的噪声所导致的不确定属性 Peng（2019）；He和Shao（2009）。为了找到可靠且有用的问题解，必须在优化中考虑这类不确定性。机会约束是对问题中的不确定性进行建模的一种自然方式。它们能够捕捉不确定变量对不等式约束的影响，并在不确定性条件下构建优化问题 Peng（2019）；Neumann 等人（2020）；Doerr 等人（2020）。机会约束通常是对问题中某个随机变量施加的不等式约束，在优化该问题时，该约束可以以极小的概率（小概率）被违反 Neumann 和 Sutton（2019）；Neumann 和 Neumann（2020）。与采用确定性方法所得的解不同，机会约束使人们能够在实施解时设定获得最大利润所需的置信水平。对具有机会约束的问题的研究有助于开发高效应用以解决复杂优化问题，并且对于为复杂问题开发现实世界解至关重要，例如采矿业 Xie 等人（2021a）；Mohtasham 等人（2021）、电力系统 Geng 和 Xie（2019）、通信系统 Abe 等人（2020）以及交通运输 Kepaptsoglou 等人（2015）。

本文研究了经典背包问题（KP）的一个变体，其中物品具有随机利润和确定性重量。由于物品的重量是确定性的，重量约束与经典背包问题中保持相同。为捕捉利润变量的不确定性，该问题引入了一个额外约束，即关于利润的机会约束 Neumann 等人（2022）。该约束保证该解具有最大利润 P，并且仅以很小的概率（α）低于 P。总之，该问题的优化过程是寻找使利润 P 最大化的解，同时满足重量约束和利润机会约束；该约束保证以至少 α 的概率获得利润 P。

在相关文献中，带有重量机会约束的背包问题（即元素具有确定性利润和随机重量）更为常见，见 Xie等人（2019、2020）。相比之下，利润机会约束背包问题仅在一项研究中得到探讨，该研究发表于近期文献中，见 Neumann 等人（2022）。该研究采用单目标进化方法来解决这一问题，见 Neumann 等人（2022）。这一情景使人们能够通过针对该问题实施特定解，识别无法达到期望利润的风险，从而在规划过程中做出更好的决策。正如 Neumann 等人（2022）所述，该问题反映了一个有益且有效的现实世界问题场景，例如矿山规划。

进化算法（进化算法）在解决随机优化问题方面表现良好，见 Doerr和Neumann（2021）；Singh和Branke（2022）。此外，对于机会约束背包问题和车间调度问题等复杂的组合优化问题，它们能够在合理的时间内生成最优解或可行解。根据底层问题的定义，进化算法可以考虑不同数量的目标。通常，当进化算法考虑单个目标时，它会生成一个使目标值达到最优的单一解。相比之下，多目标进化算法会生成一组解，在给定目标之间提供权衡。与仅得到单一解相比，这样的解集能够使人们更深入地了解如何改进算法和搜索空间，见 Coello等人（2013）；Deb（2001）。因此，多目标算法有助于人们在选择要实施的解时做出明智的决策。

本文探讨了将多目标进化方法用于 Neumann 等人（2022）提出的、具有随机利润的机会约束背包问题。本文介绍了用于解决这一问题的进化算法多目标适应度评估方法，以及提升其性能的方法。

1.1 相关工作

早期文献中已出现将进化计算用于机会约束问题的研究：He和Shao（2009）；Loughlin和Ranjithan（1999）；Liu等人（2013）；Masutomi等人（2013）。这些研究采用仿真和采样等计算开销较大的方法来处理机会约束。近期研究则转向尾界不等式，以更高效地处理机会约束：Neumann 等人（2022）；Xie等人（2019）；Assimi等人（2020）。
多篇论文研究了利润确定、重量随机的机会约束背包问题：Xie等人（2019、2020）。Xie等人（2019）介绍了如何使用广为人知的偏差不等式——切比雪夫不等式和霍夫丁界——来估计约束违反的概率。Xie等人（2020）研究了同一种背包问题变体，并为进化算法引入了适用于单目标和多目标形式的专用算子。Assimi等人（2020）研究了具有随机利润的动态机会约束背包问题。除了针对给定随机解利润的目标函数外，研究还引入了第二个目标，以处理动态容量约束。该目标刻画了满足机会约束的解所对应的最小容量界。

运行时间分析是研究机会约束问题的重要课题。首篇针对机会约束问题进行运行时间分析的论文研究了具有随机重量的背包问题 Neumann 和 Sutton (2019)。该研究考察了该问题的不同情形，并分析了 (1+1) EA 在这些情形下的运行时间。Xie 等人（2021b）对具有均匀权重的机会约束背包问题中的简单进化算法进行了运行时间分析。Neumann 和 Witt (2022) 以及 Shi 等人 (2022) 研究了简单进化算法在不同机会约束问题中的运行时间。Neumann 和 Witt (2022) 中，作者考察了具有正态分布的随机问题变量的机会约束问题中的单目标和多目标进化算法。他们还展示了如何将所提出的进化方法用于机会约束最小生成树问题 Neumann 和 Witt (2022)。Shi 等人 (2022) 分析了随机局部搜索和 (1+1) EA 在机会约束最大完工时间问题中的运行时间。

在 Neumann 等人（2022年）的研究中，作者将简单进化算法应用于利润机会约束背包问题。这些算法包括采用标准位翻转变异和重尾变异算子的（1+1）EA，以及基于种群的（μ+1）EA，后者使用了针对背包问题的特定交叉算子。本研究使用单目标适应度评估来评估所有这些算法的性能。总体结果表明，采用重尾变异算子的（1+1）EA 显著优于其他算法。

本研究的动机来自 Neumann 和 Witt（2022年）近期关于进化多目标算法的研究，该研究计算了解在期望值与方差方面的权衡。我们在此基于该研究，探讨利润机会约束背包问题的多目标形式。除了考察解的方差外，我们还考虑标准差，以及解中元素的数量（在特定条件下），以此为该问题引入适应度评估。这些适应度函数的重要之处在于，它们能够评估解的适应度，而不受其利润的任何特定置信水平影响（即不受 α 的特定值影响）。由于这会针对每种适应度形式和每种算法生成一组解，为特定目标之间提供权衡，因此有助于人们在选择要实施的解时做出更明智的决策。例如，要找出在特定 α 值下利润最高的解，我们可以计算所有解在该值下的利润，并从最终种群中选出利润最高的解。

本研究考察两种著名的多目标进化算法：Giel（2003）的 GSEMO 和 Deb 等（2002）的 NSGA-II。此外，我们为 GSEMO 引入了一种过滤方法，旨在提高该算法产生的最终种群的质量。在进化过程中，每经过固定次数的评估，便定期应用此过滤方法。该方法会根据当时的中间种群，判断某个解是否可能成为任意 $\alpha \in [0.0, 1/2]$ 的最佳解；若不能，则将其从种群中移除。在所有实验设置中，除了 GSEMO 和 NSGA-II 这两种算法外，我们还将结合此过滤方法的 GSEMO 作为一种单独的算法进行考察。

本文结构如下。第2节介绍问题形式化，第2节讨论本研究考察的算法。第3节讨论多目标形式化，包括利润目标函数、适应度函数，以及如何利用概率界估计解的置信水平。第4节和第5节讨论本文考察的进化算法及本研究提出的新过滤方法。第6节介绍实验设置的详细信息和实验结果。最后，第7节总结本研究。

2 问题定义

在经典背包问题中，物品的利润值和重量值是确定的。设背包问题有 n 个物品$\{x_1, \ldots, x_i, \ldots, x_n\}$，其利润为 p_i、重量为 w_i，重量上限为 B。问题的一个解可表示为$\{0,1\}^n$，仅当 x_i 被选入该解时，$x_i = 1$。设解 x 的利润为 $p(x) = \sum_{i=1}^{n} p_i.x_i$，重量为 $w(x) = \sum_{i=1}^{n} w_i.x_i$，则经典背包问题旨在找到解 x^*，使 $p(x^*)$最大化，并满足重量约束 $w(x^*) \leq B$。

本文研究经典背包问题的一个随机版本，其中背包物品的利润是随机的，而重量保持确定。因此，解的利润具有不确定性，可能偏离预期的最大利润值。本文采用利润机会约束来刻画问题的随机性。该约束确保对于每个可行解 x，利润低于利润值 P 的概率至多为一个很小的概率 $0 < \alpha < 1/2$。

我们可以将该问题形式化表述如下：

$$\max P \tag{1}$$
$$\text{s. t.} \quad Pr(p(x) < P) \leq \alpha \tag{2}$$
$$and \qquad w(x) \leq B \tag{3}$$

其中，α 是用于确定允许的失效概率的参数，该概率是一个较小的值 $<= 1/2$。公式 2 规定了关于利润的机会约束。

2.1 解的利润估计

一般而言，计算违反给定机会约束的概率是难以处理的 Doerr和Neumann（2021）。因此，在存在不确定性的情况下，估计一个解的利润并不简单。然而，可以使用尾界来对机会约束违反的概率进行上界估计 Doerr和Neumann（2021）。在 Neumann 等人（2022）中，作者基于以下尾界为该问题给出了利润估计值：切比雪夫不等式和霍夫丁界。这些方法中的每一种都需在特定条件下应用于该问题。

如果已知某个问题实例中解的利润期望和方差，我们可以使用切比雪夫不等式计算该问题实例中解的利润 Doerr (2020)。Neumann 等人（2022）在公式 4 中给出了利润估计值，以确保满足利润机会约束 $Pr(p(x) < P) \leq \alpha$。我们可以保证该解

x 将以较小的概率 α 以外的情况给出利润 $\hat{p}_{Cheb}(x,\alpha)$，如下所示：

$$\hat{p}_{Cheb}(x,\alpha) = \mu(x) - \sqrt{(1-\alpha)/\alpha} \cdot \sqrt{v(x)}. \tag{4}$$

上述方程给出了非常通用的解的利润设定，可用于任何已知期望值和方差的情形。例如，我们可以设定利润 p_i 服从均匀分布，使得 $p_i \in \{\mu_i - \delta, \mu_i + \delta\}$，由此得到期望值 μ_i 和方差 $\frac{\delta^2}{3}$.$|x|_1$。

如果利润是从相互独立的均匀随机分布中随机取得的，则可使用霍夫丁界来构造利润估计（Doerr，2020）。根据 Neumann 等人（2022），使用霍夫丁界（$\hat{p}_{Hoef}(x,\alpha)$）可得到解 x 的利润形式如下：

$$\hat{p}_{Hoef}(x,\alpha) = \mu(x) - \delta \cdot \sqrt{\ln(1/\alpha) \cdot 2 \cdot |x|_1}. \tag{5}$$

3 多目标形式化

本节介绍利润机会约束背包问题的多目标形式。如第2节所述，背包问题的最优解是在满足重量上界约束的条件下最大化利润。对于重量确定的利润机会约束背包问题，多目标形式需要考虑背包物品利润的不确定性。在以下小节中，我们将介绍多目标形式以及用于估计利润的函数。

3.1 适应度函数

一般而言，为了最大化解的利润，我们需要确保利润期望最大化，同时方差最小化。基于此，我们引入适应度函数 $g(x) = (\mu(x), v(x))$，该函数将生成一组在两个目标之间实现权衡的解，而不考虑 α 对单个解的利润值的保证。两个目标的公式见公式 6 和 7，其中 $v_{max} = \sum_{i=1}^{n} v_i$。

$$\mu(x) = \begin{cases} \sum_{i=1}^{n} \mu_i x_i & w(x) \leq B \\ B - w(x) & w(x) > B \end{cases} \tag{6}$$

$$v(x) = \begin{cases} \sum_{i=1}^{n} \sigma_i^2 x_i & w(x) \leq B \\ v_{max} + (w(x) - B) & w(x) > B \end{cases} \tag{7}$$

评估解的适应度时，我们需要根据不同目标确定它们之间的支配关系。对于两个解 A 和 B，若 A 支配 B（记为 $A \succeq B$），当且仅当 $\mu(A) \geq \mu(B) \wedge v(A) \leq v(B)$，且 A强支配 B（记为 $A \succ B$），当且仅当 $A \succeq B$ 且 $\mu(A) > \mu(B) \vee v(A) < v(B)$。对于违反重量容量约束（$w(A) > B$）的不可行解，上述公式会对两个目标值进行惩罚（见公式 6 和 7）。这种形式确保任何可行解都支配不可行解实例。
接下来，我们考虑适应度函数 $g'(x) = (\mu(x), s(x))$。该适应度函数仅第二个目标与 $g'(x)$ 不同，而第一个目标 μ 保持不变，可按公式 6 定义。我们将公式 8 中的最大标准差记为 $s_{max} = \sum_{i=1}^{n} \sigma_i$，该公式定义了此适应度函数的第二个目标 $g'(x)$。

$$s(x) = \begin{cases} \sqrt{\sum_{i=1}^{n} \sigma_i^2 x_i} & w(x) \leq B \\ s_{max} + (w(x) - B) & w(x) > B. \end{cases} \tag{8}$$

对于 GSEMO 这类简单进化算法，优化 $g(x)$ 时的最终种群与优化 $g'(x)$ 时的最终种群等价，因为两者的区别在于，后者的第二个目标是前者第二个目标的平方根。因此，这不会改变此类算法搜索点的顺序。不过需要注意的是，一些流行的进化多目标优化算法（如 NSGA-II）在处理 $v(x)$ 而非 $s(x)$ 时，其多样性机制会有所不同。因此，本文同时研究 $g(x)$ 和 $g'(x)$。

3.2 相同离散度下利润的适应度函数

当所有元素的 δ 值相同时，我们可以将解中所选物品的数量（$|x|_1$）视为第二个目标。这个目标使我们能够定义一个适应度函数，该函数不受解的置信水平（α）以及单个利润值的不确定性水平（δ）的影响。我们将此情形下的适应度函数记为 $g''(x) = (\mu(x), c(x))$。新适应度函数的第一个目标基于 x $\mu(x)$ 的利润期望，与之前的函数类似，并按公式 6 所示进行计算。第二个目标 $c(x)$的计算如下：

$$c(x) = \begin{cases} \sum_{i=1}^{n} x_i & w(x) \leq B \\ n + (w(x) - B) & w(x) > B. \end{cases} \tag{9}$$

3.3 最佳利润值的估计

作为多目标进化方法的最终输出，我们得到一组解，这些解对目标之间进行了权衡，且与利润的置信水平无关。我们需要利用这些最终解，找出给定置信水平 α 下利润最高的解。我们可以使用 Neumann 等人（2022）的公式 4 和 5，计算最终种群中所有解在所需置信水平（α）下的利润。对于每个 α，利润值最高的解就是该特定设置下的最佳解。

3.4 解的利润置信水平

多目标方法产生的最终种群中的不同解，会在不同置信水平下成为最佳解。我们可以确定最终种群中每个解在哪个置信水平下能取得最高利润值。首先，通过比较第 2.1 小节中提到的利润估计值，我们得到一对解之间的 α 阈值，使其中一个解优于另一个解。根据最终种群中解对之间的阈值 α 值，我们为每个解定义一个置信水平区间，使其在该区间内的任意 α 下都能取得最高利润值。

3.4.1 使用切比雪夫不等式的置信水平阈值

设解 x 和 y 满足 $\mu(x) > \mu(y)$ 且 $v(x) > v(y)$，则当最低要求的置信水平（$\alpha_{x,\ y}$）如下时，我们可以说，解 x 的利润优于 y：

$$\alpha_{Cheb(x,y)} = \frac{1}{1 + (R(x,y))^2} \text{ s.t. } \qquad R(x,y) = \frac{\mu(x) - \mu(y)}{\sqrt{v(x)} - \sqrt{v(y)}} \tag{10}$$

对于置信 水平 $\alpha \geq 1/(1 + (R(x,y))^2)$，解 X 的利润值将优于解 Y .

定理 3.1。设 $0 < \alpha < 1$、$and\ x$ 和 y 为两个可行解，满足 $\mu(x) > \mu(y)$ 且 $v(x) > v(y)$。若 $\alpha \geq {}_{1+\ R}{}^{1}{}_{((x,\ y))}$ ^{2} 成立，~~且满足 $R(x,y) = {}^{\mu}{}_{(x)-\mu(y)\sqrt{v(x)}-\sqrt{v(y)}}$~~，则 $\hat{}p_{Cheb}(x,\alpha) \geq \hat{}p_{Cheb}(y,\alpha)$。

证明。我们有

$$\begin{aligned} & \alpha \geq \frac{1}{1 + (R(x,y))^2} \\ \iff \quad & \alpha + \alpha(R(x,y))^2 \geq 1 \\ \iff \quad & (R(x,y))^2 \geq (1-\alpha)/\alpha \end{aligned}$$

由于我们假设 $0 < \alpha < 1, \mu(x) > \mu(y)$ 且 $v(x) > v(y)$，因此有 $R(x,y) > 0$ 和 $\frac{1-\alpha}{\alpha} > 0$。这意味着，

$$\begin{aligned} & R(x,y) \geq \sqrt{(1-\alpha)/\alpha} \\ \iff \quad & \frac{\mu(x)-\mu(y)}{\sqrt{v(x)}-\sqrt{v(y)}} \geq \sqrt{(1-\alpha)/\alpha} \\ \iff \quad & \mu(x)-\mu(y) \geq \sqrt{\frac{1-\alpha}{\alpha}} \cdot \left(\sqrt{v(x)}-\sqrt{v(y)}\right) \\ \iff \quad & \mu(x)-\sqrt{\frac{1-\alpha}{\alpha}} \cdot \sqrt{v(x)} \geq \mu(y)-\sqrt{\frac{1-\alpha}{\alpha}} \cdot \sqrt{v(y)} \\ \iff \quad & \hat{p}_{Cheb}(x,\alpha) \geq \hat{p}_{Cheb}(y,\alpha) \end{aligned}$$

□

3.4.2 使用Hoeffding界的置信水平阈值

考虑解 x 和 y，使得 $\mu(x) > \mu(y)$ 且 $v(x) > v(y)$。根据 $\hat{p}_{Hoef}$，我们可以推导出最小置信水平 $\alpha_{x,\ y}$，使得相对于 $\hat{p}_{Hoef}$，解 x 的利润优于 y，如下所示：

$$\alpha_{Hoef(x,y)} = e^{-(S(x,y))^2} \text{ s.t.} \qquad S(x,y) = \frac{\mu(x)-\mu(y)}{\delta\left(\sqrt{2|x|_1}-\sqrt{2|y|_1}\right)} \tag{11}$$

定理3.2。设 $0 < \alpha < 1$、$and\ x$ 和 y 为两个可行解，使得 $\mu(x) > \mu(y)$ 且 $v(x) > v(y)$ 成立。如果 $\alpha \geq e^{-(S(x,y))^2}$ 成立，则 $\hat{p}_{Hoef}(x,\alpha) \geq \hat{p}_{Hoef}(y,\alpha)$。

证明。我们有，

$$\begin{aligned} & \alpha \geq e^{-(S(x,y))^2} \\ \iff \quad & e^{(S(x,y))^2} \geq 1/\alpha \\ \iff \quad & (S(x,y))^2 \geq \ln(1/\alpha) \end{aligned}$$

由于我们假设 $0 < \alpha < 1$、$\mu(x) > \mu(y)$ 且 $v(x) > v(y)$，因此有 $S(x,y) > 0$ 以及 $\ln\ \frac{1}{\alpha} > 0$。这意味着，

$$\begin{aligned} & S(x,y) \geq \sqrt{\ln(1/\alpha)} \\ \iff \quad & \frac{\mu(x)-\mu(y)}{\delta\left(\sqrt{2|x|_1}-\sqrt{2|y|_1}\right)} \geq \sqrt{\ln(1/\alpha)} \\ \iff \quad & \mu(x)-\mu(y) \geq \delta \cdot \sqrt{2\ln\left(\frac{1}{\alpha}\right)}\left(\sqrt{|x|_1}-\sqrt{|y|_1}\right) \\ \iff \quad & \mu(x)-\delta \cdot \sqrt{\ln\left(\frac{1}{\alpha}\right) \cdot 2 \cdot |x|_1} \geq \mu(y)-\delta \cdot \sqrt{\ln\left(\frac{1}{\alpha}\right) \cdot 2 \cdot |y|_1} \\ \iff \quad & \hat{p}_{Hoef}(x,\alpha) \geq \hat{p}_{Hoef}(y,\alpha) \end{aligned}$$

□

3.4.3 最终种群中解的置信水平区间

我们可以使用前面各小节中的 α 阈值，来引入种群中解的置信水平范围，具体如下。作为 α 阈值，我们可以使用 $\alpha_{Cheb(x,y)}$ 或 $\alpha_{Hoef(x,y)}$。无论用于估计 α 阈值的具体公式是什么，我们都可以为一个解引入 α 区间。

首先，我们将给定种群 P 中的所有解排序为 $\{x_1, ...x_n\}$，使得 $\mu(x_1) \geq \mu(x_2) \geq ... \geq \mu(x_n)$。然后，我们可以定义一个 $(n+1) \times (n+1)$ 的置信水平阈值对称矩阵，如下所示：

算法1 GSEMO

1：均匀随机选择 $x \in \{0,1\}^n$；2：　$S \leftarrow \{x\}$；3：当未满足停止条件时执行4：均匀随机选择 $x \in S$；5：$y \leftarrow$ 以概率 $\frac{1}{n}$ 独立地翻转 x 的每一位；6：如果（$\nexists w \in S$：$w \succ y$）则7：$S \leftarrow (S \cup \{y\}) \backslash \{z \in S \mid y \succeq z\}$；8：结束 if9：结束 while

$$\alpha_{(i,j)} = \begin{cases} 1 & i = 0 \text{ or } j = 0 \\ \alpha_{Cheb(i,j)} \text{ or } \alpha_{Hoef(i,j)} & i = 1, ..., n \end{cases} \tag{12}$$

最后，利用置信水平阈值，可以得到解 k 的置信水平范围，如公式13所示。如果某个解存在有效的 α 区间，那么在该区间内的 α 值下，该解将取得最高利润值。如果某个解的 α 区间为空，那么该解在任何 α 值下都不会是最优解。

$$\min_{j=0}^{k-1} \alpha_{i,k} \leq \alpha_k \leq \max_{j=k}^{n-1} \alpha_{i,j} \tag{13}$$

4 算法

本研究考虑两种广泛使用的多目标进化算法：GSEMO 和 NSGA-II。GSEMO 是用于解决多目标优化问题的最基础进化算法。许多研究已证明，它在解决机会约束多目标优化问题方面十分有效（Neumann 等人（2022））。GSEMO 的步骤见算法1。

种群 S 最初包含一个随机生成的解。在迭代过程中，从 S 中均匀随机地选择一个父代解 x，并以$\frac{1}{n}$的概率翻转 x的每一位，从而生成一个子代解 y。如果 y不被 S 中任何现有解支配，则将其加入 S，并替换 S 中所有被 y 支配的现有解。这样可以确保每次迭代结束时，种群 S 都包含一组相对于给定目标函数同样有效的非支配解。

NSGA-II 是最著名的多目标进化算法之一，其通过寻找近最优解来提升多样性。Deb等（2002）。如果我们考虑 NSGA-II 进化过程的一次迭代，它会根据父代种群创建一个子代种群。首先，我们使用二元锦标赛选择两个父代解，并应用单点交叉生成两个子代解；随后，通过以 $\frac{1}{n}$ 的概率翻转每一位对其进行变异。当子代种群已满时，将父代解和子代解中的所有解放在一起考虑，并将其划分为非支配解前沿（即，一个前沿中的解不会支配其中的其他解）。从等级最低的前沿开始，依次将其加入新种群，直到达到容量上限。如果只需将某个前沿的一部分纳入新种群，则使用基于拥挤距离的排序来决定选择哪些解。这有助于使解分布得更加均匀（提升多样性）。在该算法的一次迭代中，需要进行一次新的适应度评估，其数量等于子代种群的规模。算法的迭代次数取决于子代种群规模以及实验设置中所考虑的最大评估次数。

5 机会约束问题的过滤

这里，我们为 GSEMO 提出一种过滤方法，通过从中间种群中移除没有有效α 区间的解，来改进其结果。GSEMO 的最终种群中包含一些解，对于 α 的任何概率值，这些解都无法取得最佳利润值。此类解对于在给定置信水平下寻找最佳解这一优化目标毫无助益。例如，图1展示了 GSEMO 在 uncorr-100 上使用 1000 万次适应度评估得到的最终种群；uncorr-100 是实验中使用的一个实例。该图展示了最终种群中所有解的 $\mu(x)$ 与$v(x)$ 的关系。蓝色星形标记表示该解具有有效的置信水平区间。这些解的 α 区间构成了完整的概率范围 [0.0,1.0]。在优化结束时

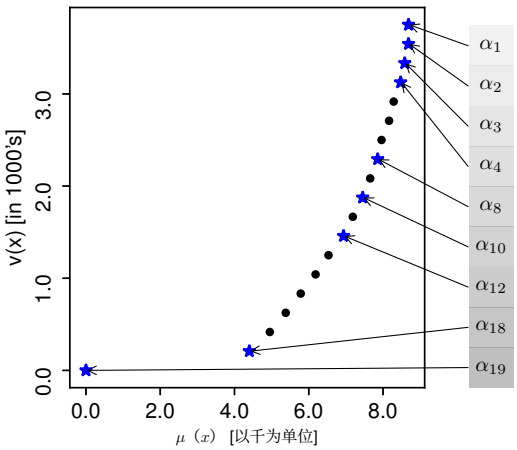

图1：GSEMO 的最终种群示例

算法2 过滤方法

0: 输入：种群 $P_0 = x^1, ..., x^n$ 按 $\mu(x^i)$ 递减顺序排列；1: 设置 $k \leftarrow 1$；2: 当 $k \leq n$时执行3: 设置 $upper \leftarrow \min_{j=0}^{k-1} \alpha_{i,k}$4: 设置 $lower \leftarrow \max_{j=k}^{n-1} \alpha_{i,j}$5: 如果 $upper \geq lower$ 则6: $P_1 \cup \{x^k\}$7: 结束如果8: 设置 $k \leftarrow k + 1$9: 结束循环10: 返回 P_1

在这一过程中，我们关注带有这些蓝色标记的解。其中一个解会成为给定 α 值下的最佳解。相反，其他以黑色标记的解在任何置信水平下都不会成为最佳解。

过滤方法会移除这些在任何置信水平下都不会成为最佳解的解。经过一定次数的适应度评估后，该方法会定期应用于 GSEMO 的中间种群。此过程会根据公式13所示的 α 区间表示，从中间种群中移除相应的解。由于过滤方法只保留具有有效 α 区间的解，因此进化优化有更大的机会改进这些解。因此，该方法有助于提高最终种群中解的质量。

过滤方法的步骤如算法2所示。它以种群 P_0 作为输入，该种群可以是最终种群，也可以是在 GSEMO 执行过程中生成的中间种群。种群 P_0 中的解需要按照 $\mu(x)$降序排列。对于每个解 x^k，我们考虑$\alpha_{i,\ k}$ 并使用公式13中的不等式选择置信水平区间。若α_k 的区间非空，则将解 x^k 添加到结果种群 P_1 中。

6 实验

在本研究中，我们使用多目标进化算法，在不同的基准测试上针对不同的机会约束设置评估前文介绍的适应度函数，并通过多项实验开展评估。本文将介绍实验设置及实验结果。

ith $g''(x)$ (5)	
std	stat
2.5740	$1^{-}2^{-}$
0.1309	$1^{-}2^{-}$
6.9209	$1^{-}2^{-}$
3.8802	$1^{-}2^{-}$
9.5604	$1^{-}2^{-}3^{-}$
9.5267	$1^{-}2^{-}3^{-}$
4.5506	2^{-}
5.3753	2^{-}
3.3765	2^{-}
3.8490	2^{-}
2.0052	2^{-}
1.0454	$2^{+}3^{-}4^{-}$
9.0223	2^{-}
3.5459	2^{-}
6.3437	2^{-}
3.2449	2^{-}
7.0386	2^{-}
0.2576	2^{-}
2.7994	1^{+}
3.8869	$1^{+}2^{-}$
2.1891	1^{+}
2.9564	$1^{+}2^{-}$
9.2069	1^{+}
3.3870	1^{+}
7.2936	2^{-}
3.2837	$1^{-}2^{-}$
6.4267	2^{-}
6.5882	$1^{-}2^{-}$
1.5375	2^{-}
7.2660	2^{-}
2.0549	$1^{+}2^{-}$
2.3801	$1^{+}2^{-}$
9.6260	$1^{+}2^{-}$
7.2743	$1^{+}2^{-}$
9.3640	$1^{+}2^{-}$
1.6241	1^{+}

6.1 实验设置

在实验研究中，我们采用与 Neumann 等人（2022）相同的基准测试。这组基准测试包含三个相关实例和三个有界强相关实例，每个实例包含 $n \in \{100, 300, 500\}$ 个背包物品。我们假设背包物品的利润服从均匀分布，因此元素 i 的利润从 $p_i \in \{\mu_i - \delta, \mu_i + \delta\}$ 中均匀随机选取。这样，当利润具有相同的不确定性水平（δ）时，便可以同时使用 $\hat{}p_{Cheb}$ 和 $\hat{}p_{Hoef}$。对于每个基准测试，实验研究涵盖两个不确定性水平： $\delta \in \{25, 50\}$。此外，我们还考虑利润具有不同离散度的情形，此时每个物品 i 的不确定性水平为 δ_i，并从 $\delta_i \in [0.0, \mu_i]$ 中均匀随机选取。仅针对 $\hat{}p_{Cheb}$ 考虑不确定性各异的基准测试，因为 $\hat{}p_{Hoef}$ 要求所有元素具有相同的不确定性水平。

本研究主要考察三种算法：两种知名的多目标进化算法 GSEMO 和 NSGA-II，以及采用第4节所介绍过滤方法的 GSEMO。本文后文将后者称为 GSEMO+Filtering。我们根据需要将这些算法与适应度函数结合，用于实验。对于具有固定不确定性的基准测试，使用 GSEMO 或 GSEMO+Filtering 搭配任意适应度函数（$g(x)$ 或 $g'(x)$ 或 $g''(x)$）时，最终种群将产生等效结果。因此，GSEMO 和 GSEMO+Filtering 仅考虑采用适应度评估 $g(x)$。当利润具有相同离散度时，适应度函数 $g''(x)$ 会考量解中所选物品的数量。因此，对于利润具有不同不确定性的基准测试，我们不考虑采用该适应度函数的算法。对于这些基准测试，仅考虑采用 $g(x)$ 的 GSEMO 和 GSEMO+Filtering，以及采用 $g(x)$ 和 $g'(x)$ 的 NSGA-II。
每种算法都进行 1000 万次适应度评估，并生成一个解种群，在适应度评估所采用的目标之间进行权衡。这些方法的输出质量通过分析不同置信水平下的最佳利润值来评估，其中考虑了 $\alpha \in \{0.1, 0.01, 0.001\}$。每种方法都会生成一个独立于 α 的最终种群，并针对不同的 α 值，使用相应的利润估计 $\hat{}p_{Cheb}$ 或 $\hat{}p_{Hoef}$，从该种群中选出最佳解。结果汇总了每个 α 的 30 次实验所得的最佳利润值。对于利润离散度相同的基准测试，此汇总要求针对每个 δ 运行该方法 30 次；对于利润离散度不同的基准测试，也需运行 30 次。不过，使用 $g''(x)$ 时，由于算法的运行独立于 δ 值，因此可以从同一个最终种群中获得不同 δ 对应的最佳利润值。

最后，我们使用置信水平为 95% 的 Kruskal-Wallis检验，并结合 Bonferroni事后统计程序，检验结果在统计显著性方面是否有效。统计比较标记为 X^+ 或 X^-，表示该列中的方法优于 X，反之亦然。如果两种方法之间没有显著差异，则不显示相应数字。对于每种方法，最佳利润值的汇总以均值、标准差和统计量表示，分别代表结果的均值和标准差，以及与其他方法对应结果的统计比较。

6.2 结果

表1和表2给出了元素利润具有固定不确定性水平的基准测试结果。根据均值，在两张表的大多数设置中，采用过滤方法的GSEMO优于其他方法。统计比较进一步揭示了各方法应用于每个实例时的性能。表1中uncorr-100的结果显示，GSEMO+过滤的表现优于其他四种方法，而GSEMO优于采用 $g''(x)$的NSGA-II。对于较小的置信水平 $\alpha = 0.001$，采用 $g(x)$的NSGA-II优于采用 $g''(x)$的NSGA-II。对于 $\alpha = 0.001$ 和 $\delta = 50$，strong-100实例的结果与同一实例的其他情况大不相同。在这些情况下，采用 $g(x)$的GSEMO以及采用 $g(x)$和 $g'(x)$的NSGA-II表现良好，而GSEMO+过滤取得了最低结果。然而，第二种方法在其他 α 和 δ 设置下表现良好，采用 $g(x)$和 $g'(x)$的NSGA-II在某些设置下也表现相似。

对于 uncorr-300 的所有设置，GSEMO+过滤的表现都优于其他四种方法，而其他方法彼此表现相近。对于 strong-300 实例，GSEMO 的结果低于 GSEMO+过滤和 NSGA-II（ $g(x)$ 和 $g''(x)$ ）。此外，对于 $\alpha = 0.001$，采用 $g(x)$ 的 NSGA-II 结果与 GSEMO+过滤一样好。对于 uncorr-500，GSEMO+过滤取得了最佳结果，且 GSEMO 的表现优于大多数 NSGA-II 结果。不过，在该实例上使用 NSGA-II 和 $g''(x)$ 时，除置信水平为 $\alpha = \{0.1, 0.01\}$ 外，结果表现相同；此处考察的不确定性水平为 $\delta = 50$。strong-500 上的实验同样显示，GSEMO+过滤取得了最佳结果。然而，对于该实例所考察的所有设置，其他方法的表现都优于 GSEMO。相比之下，NSGA-II 在各设置下均取得第二好的结果；当 $\delta = 50$ 时， $\alpha = 0.001$NSGA-II 方法的表现也与 GSEMO+过滤相当。

表2：使用霍夫丁界时相同离散度的结果

	B	α	δ	使用 g(x) 的 GSEMO (1)			GSEMO+过滤，使用 g(x) (2)			使用 g(x) 的 NSGA-II (3)			使用 g′(x) 的 NSGA-II (4)			NSGA-II 与 g′′(x) (5)		
				mean	std	stat	mean	std	stat	mean	std	stat	mean	std	stat	mean	std	stat
uncorr-100	2407	0.1	25	10987.3004	75.7683	2^-	**11042.7989**	0.0000	$1^+3^+4^+5^+$	10965.6291	88.0780	2^-	10956.9290	50.1020	2^-	10965.9887	51.9898	2^-
			50	10778.0078	60.1974	2^-	**10821.5978**	0.0000	$1^+3^+4^+5^+$	10753.4968	51.9386	2^-	10728.1263	79.0134	2^-	10749.4108	48.9647	2^-
		0.01	25	10896.5878	74.5928	2^-	**10951.1744**	0.0000	$1^+3^+4^+5^+$	10875.7430	86.4446	2^-	10867.4019	48.9713	2^-	10876.2792	50.7360	2^-
			50	10596.7649	57.6057	2^-	**10638.3488**	0.0000	$1^+3^+4^+5^+$	10573.7130	49.3723	2^-	10549.2659	76.1316	2^-	10569.9917	46.4640	2^-
		0.001	25	10826.9815	73.6938	2^-	**10880.8684**	0.0000	$1^+3^+4^+5^+$	10806.7709	85.1930	2^-	10798.7053	48.1051	2^-	10807.4426	49.7746	2^-
			50	10457.6924	55.6226	2^-	**10497.7369**	0.0000	$1^+3^+4^+5^+$	10435.7601	47.4116	2^-	10412.0216	73.9290	2^-	10432.3187	44.5489	2^-
strong-100	4187	0.1	25	8463.1467	127.6172	2^-	**8561.1573**	85.4504	$1^+3^+4^+5^+$	8481.7278	123.3848	2^-	8456.8099	146.8670	2^-	8456.3390	122.3538	2^-
			50	8278.6327	90.2553	2^-	**8332.4692**	64.4783	$1^+3^+5^+$	8240.4850	110.0787	2^-	8276.2257	119.9424		8233.2780	111.0076	2^-
		0.01	25	8369.3409	122.2698	2^-	**8464.4621**	81.5096	$1^+3^+4^+5^+$	8387.9199	118.6225	2^-	8364.1958	141.2465	2^-	8363.9441	117.6448	2^-
			50	8086.8101	82.3211	2^-	**8139.0049**	58.6173	$1^+3^+5^+$	8054.9801	100.8829	2^-	8087.7692	109.6779		8048.4883	101.6750	2^-
		0.001	25	8297.3868	118.1961	2^-	**8390.2653**	78.4972	$1^+3^+4^+5^+$	8315.9386	114.9756	2^-	8293.1306	136.9396	2^-	8293.0470	114.0400	2^-
			50	7939.6195	76.3766	2^-	**7990.5545**	54.1630	$1^+3^+5^+$	7912.6372	93.8879	2^-	7943.1615	101.8268		7906.6941	94.5672	2^-
uncorr-300	6853	0.1	25	33863.1544	205.0835	2^-	34212.8177	146.9244	$1^+3^+4^+5^+$	33610.2955	532.9802	2^-	33600.5469	554.0346	2^-	33545.0211	487.7131	2^-
			50	33428.2570	259.2905	2^-	33821.1723	157.7349	$1^+3^+4^+5^+$	33276.1084	510.0035	2^-	33143.0191	448.4365	2^-	33177.8089	480.9303	2^-
		0.01	25	33708.5096	203.9304	2^-	34055.7967	145.6324	$1^+3^+4^+5^+$	33457.5385	530.1364	2^-	33448.1219	551.0166	2^-	33392.9168	484.9026	2^-
			50	33119.8295	255.9406	2^-	33507.4493	154.5170	$1^+3^+4^+5^+$	32970.6017	504.5017	2^-	32839.2289	443.7443	2^-	32873.6003	475.3176	2^-
		0.001	25	33589.8465	203.0502	2^-	33935.3102	144.6467	$1^+3^+4^+5^+$	33340.3278	527.9571	2^-	33331.1621	548.7016	2^-	33276.2031	482.7469	2^-
			50	32883.1649	253.3855	2^-	33266.7212	152.0660	$1^+3^+4^+5^+$	32736.1783	500.2836	2^-	32606.1226	440.1485	2^-	32640.1729	471.0147	2^-
strong-300	13821	0.1	25	23744.0892	430.4747	$2^-3^-5^-$	24300.5479	214.9589	$1^+3^+4^+$	24033.7421	324.1660	1^+2^-	23921.4833	341.5510	2^-	24109.9465	230.6676	1^+
			50	23462.9510	329.9513	$2^-3^-5^-$	23997.1125	215.9394	$1^+3^+4^+5^+$	23735.7246	288.1238	1^+2^-	23566.2352	299.2535	2^-	23767.0263	219.6354	1^+2^-
		0.01	25	23603.7492	424.5461	$2^-3^-5^-$	24152.7138	210.6726	$1^+3^+4^+$	23892.9269	317.9729	1^+2^-	23782.5354	335.8005	2^-	23967.9043	226.0936	1^+
			50	23181.2241	317.9412	$2^-3^-5^-$	23700.6927	206.3054	$1^+3^+4^+5^+$	23452.1877	276.3622	1^+2^-	23289.4860	288.4378	2^-	23482.9419	210.5250	1^+2^-
		0.001	25	23496.0973	420.0184	$2^-3^-5^-$	24039.3181	207.2892	$1^+3^+4^+$	23784.9190	313.2317	1^+2^-	23675.9269	331.3992	2^-	23858.9115	222.5879	1^+
			50	22965.0474	308.7926	$2^-3^-5^-$	23473.2418	198.9571	$1^+3^+4^+5^+$	23234.6394	267.3758	1^+2^-	23077.1290	280.1499	2^-	23264.9563	203.5545	1^+2^-
uncorr-500	11243	0.1	25	56985.6975	747.8288	$2^-3^+4^+$	58337.8820	310.8352	$1^+3^+4^+5^+$	55761.8933	1247.6913	1^-2^-	55780.3693	1406.1906	1^-2^-	55948.9616	1285.1401	2^-
			50	56509.1689	856.1817	$2^-3^+4^+5^+$	57934.0703	312.7435	$1^+3^+4^+5^+$	55386.3718	1041.2515	1^-2^-	54806.2623	1219.9708	1^-2^-	55489.9899	1273.9787	1^-2^-
		0.01	25	56790.6294	745.4714	$2^-3^+4^+$	58137.6851	309.2459	$1^+3^+4^+5^+$	55572.3326	1243.5564	1^-2^-	55590.6468	1401.5183	1^-2^-	55758.8493	1280.5162	2^-
			50	56119.2292	849.6132	$2^-3^+4^+5^+$	57533.3999	309.3150	$1^+3^+4^+5^+$	55007.2442	1033.3396	1^-2^-	54431.0658	1211.8786	1^-2^-	55109.7653	1264.7366	1^-2^-
		0.001	25	56640.9485	743.6632	$2^-3^+4^+$	57984.0686	308.0287	$1^+3^+4^+5^+$	55426.8776	1240.3841	1^-2^-	55445.0676	1397.9335	1^-2^-	55612.9710	1276.9688	2^-
			50	55820.0179	844.5763	$2^-3^+4^+5^+$	57226.0199	306.6676	$1^+3^+4^+5^+$	54716.3294	1027.2713	1^-2^-	54143.1674	1205.6714	1^-2^-	54818.0086	1257.6477	1^-2^-
strong-500	22223	0.1	25	38739.7417	688.5044	$2^-3^-4^-5^-$	40301.3374	447.3446	$1^+3^+4^+5^+$	39707.3872	412.9829	1^+2^-	39669.4439	422.3664	1^+2^-	39685.3277	389.7418	1^+2^-
			50	38280.9153	613.7020	$2^-3^-4^-5^-$	39897.8123	344.8981	$1^+3^+4^+5^+$	39273.3775	598.3347	1^+2^-	39315.7031	477.1865	1^+2^-	39247.8886	377.7601	1^+2^-
		0.01	25	38563.3178	680.7776	$2^-3^-4^-5^-$	40109.3511	442.0526	$1^+3^+4^+5^+$	39525.9425	407.4683	1^+2^-	39488.2712	416.7806	1^+2^-	39504.1345	384.7767	1^+2^-
			50	37930.8422	599.1733	$2^-3^-4^-5^-$	39510.9695	336.7709	$1^+3^+4^+5^+$	38909.4678	582.7780	1^+2^-	38950.9755	464.0994	1^+2^-	38885.5815	367.8049	1^+2^-
		0.001	25	38427.9430	674.8592	$2^-3^-4^-5^-$	39962.0501	437.9920	$1^+3^+4^+5^+$	39386.7151	403.2421	1^+2^-	39349.2525	412.4983	1^+2^-	39365.1001	380.9688	1^+2^-
			50	37662.2216	588.0761	$2^-3^-4^-5^-$	39214.1346	330.5707	$1^+3^+4^+5^+$	38630.2301	570.8539	1^+2^-	38671.1102	454.0722	1^+2^-	38607.6080	360.1434	1^+2^-

表3：使用切比雪夫不等式时不同离散度的结果

实例	B	α	GSEMO 与 g(x)（1）			GSEMO+过滤与 g(x)（2）			NSGA-II 与 g(x)（3）			NSGA-II 与 g′(x)（4）		
			mean	std	stat	mean	std	stat	mean	std	stat	mean	std	stat
uncorr-100	2407	0.1	8546.7015	322.7956		8562.9620	323.1475		**8565.8942**	324.9223		8564.2549	323.8494	
		0.01	**4682.2710**	799.1875		4675.9295	795.5029		4674.8735	797.5334		4677.1350	799.3174	
		0.001	**1967.4965**	880.0104		1962.3962	886.0726		1919.2445	917.6798		1953.6975	894.4411	
strong-100	4187	0.1	7100.1631	245.6990		7099.8604	250.2445		7122.7698	249.2182		**7123.3386**	250.5080	
		0.01	5315.4258	326.6578		5308.5248	343.0757		5329.4856	326.9290		**5333.3263**	325.5847	
		0.001	**3746.9429**	732.1072		3746.1213	731.9397		3734.2093	731.9080		3743.5859	731.6689	
uncorr-300	6853	0.1	29089.9230	479.0749	$2^-3^-4^-$	29580.8035	355.6388	1^+	**29735.0392**	358.7213	1^+	29676.9763	388.6763	1^+
		0.01	19396.8836	1128.3967		19431.8036	1126.0478		19589.0418	1131.0819		19580.3092	1131.0342	
		0.001	8910.0087	1370.4531		8850.8254	1381.8863		8899.5025	1378.4832		8971.3050	1383.6696	
strong-300	13821	0.1	21789.9295	335.7672	$2^-3^-4^-$	22171.9138	358.9991	1^+	**22345.3397**	309.3574	1^+	22297.4300	307.0689	1^+
		0.01	18172.8378	560.3248		18195.2974	615.1645		18338.0359	588.6787		**18342.4421**	576.7652	
		0.001	14629.7944	809.3377		14617.5558	794.6289		14643.0814	808.6424		**14667.4349**	812.9751	
uncorr-500	11243	0.1	60266.4398	709.0211	$2^-3^-4^-$	**52494.0984**	556.8082	1^+	52468.0194	532.9634	1^+	52149.3408	700.4027	1^+
		0.01	37753.4240	1566.1944		38510.4882	1564.4777		38746.1887	1539.8167		38686.7230	1555.8618	
		0.001	18969.0800	2144.1783		18880.7433	2144.3506		19153.0886	2137.5579		19190.4696	2134.7447	
strong-500	22223	0.1	35919.5415	631.8822	$2^-3^-4^-$	**37833.8138**	352.1352	1^+	37832.1320	332.8651	1^+	37690.0363	317.8082	1^+
		0.01	30977.9111	679.1163	$2^-3^-4^-$	31554.3119	682.8664	1^+	**31822.0362**	649.7576	1^+	31805.6899	637.4521	1^+
		0.001	25041.2112	721.5121		25018.0126	704.1720		25131.2311	723.9024		**25193.7178**	741.8816	

表2给出了使用 $\hat{p}_{Cheb}$ 估算解的利润值的实验结果。在 uncorr-100、strong-100 和 uncorr-300 上，使用 GSEMO+过滤时结果最高。除在不确定性水平为 $\delta = 50$ 时、针对 strong-100 实例采用 $g'(x)$ 的 NSGA-II 外，它的表现优于所有其他方法。strong-300 实例上的实验表明，对于背包物品较低的不确定性水平（25），采用 $g''(x)$ 的 NSGA-II 表现与 GSEMO+过滤相当。对于该实例，GSEMO 和采用 $g'(x)$ 的 NSGA-II 方法结果最低。相反，在 uncorr-500 上，GSEMO 的表现优于大多数 NSGA-II 方法。使用 $g''(x)$ 时，在较低的不确定性水平值 25 下，NSGA-II 的表现与 GSEMO 相当。对于所有 α 和 δ 值，表2中 strong-500 上的实验显示，NSGA-II 方法的表现均不及 GSEMO+过滤。不过，在该基准测试上，NSGA-II 方法的表现优于 GSEMO。

GSEMO+过滤在应用于利润具有相同离散度的基准测试时表现显著出色。过滤方法使中间种群能够包含更多产生有效置信水平区间的解。因此，进一步改进这些解最终会带来更好的结果。考虑 NSGA-II 的结果时， $g(x)$ 和 $g'(x)$ 往往比 $g''(x)$ 产生更好的结果。这可能是因为在考虑解的利润方差或标准差时，拥挤距离分配效果更好。

我们可以将利润具有相同离散度的基准测试结果（见表1和表2）与 Neumann 等人（2022）的先前工作进行比较，因为该研究也采用了相同的基准测试和相似的实验设置。两项工作的实验设置相同，唯一的区别在于算法所进行的适应度评估次数。对于每个 α 和 δ 值，Neumann 等人（2022）的方法运行了 100 万次适应度评估。相比之下，我们针对每个 δ 值在基准测试上运行多目标方法，进行 1000 万次适应度评估，由同一算法的输出得出所有 α 值对应的结果。结果显示，在切比雪夫结果中，36 种设置有 14 种的最高均值利润值（见表1）优于 Neumann 等人（2022）使用的全部三种方法；在霍夫丁结果中，36 种设置有 15 种获得了更高利润（见表2）。总体而言，这些情况大多来自不相关基准测试上的实验。对于新方法在有界强相关实例上表现更好的情况，通常对应较高的不确定性水平 $\delta = 50$ 和较低的置信度值，例如$\alpha = \{0.01, 0.01\}$。

表3展示了不同利润离散度下各基准测试的结果。对于每种情况，报告的最高均值表明，在较小的实例中，GSEMO 和 NSGA-II 的平均利润最高；而对于包含 300 或 500 个元素的实例，GSEMO+过滤（ $g'(x)$）和 NSGA-II 的平均利润最高。尽管不同方法取得了最高均值，但在大多数设置下，结果都相近。根据统计比较可以看出，较小的实例 strong-100 和 uncorr-100 中，各方法的结果并无显著差异。其他实例的结果表明，对于 $\alpha = 0.1$ GSEMO+过滤和 NSGA-II 方法，其表现优于 GSEMO。此外，在 strong-500 实例中，GSEMO+过滤（$g'(x)$）和 NSGA-II 在 $\alpha = 0.01$ 的两种适应度函数下都取得了更好的结果。然而，在其他采用 $\alpha = \{0.01, 0.001\}$ 的设置下，针对包含 300 和 500 个物品的实例，所有方法的结果都相近。

与利润具有相同离散度的基准测试相比，离散度不同的基准测试可能会使某些元素的利润方差更高。因此，确保最终种群中的解具有良好的分布至关重要，因为在考虑特定置信水平时，更多解往往会产生负利润值。NSGA-II 在适应度评估中采用选定目标并使用基于拥挤距离的排序时，似乎能够实现这一点。相比之下，GSEMO+过滤也能取得同样出色的结果：它会忽略不具有有效 α 区间的解，从而逐步改进最终种群。

7 结论

本文探讨了用于求解利润机会约束背包问题的多目标进化方法。我们为进化算法引入了适用于该问题的适应度评估方法。这些适应度函数能够评估解，而不受其所需置信水平的影响。因此，进化算法的结果会形成一个种群，其中包含在不同置信水平下能够取得最佳利润值的解。因此，无须在运行算法之前确定所需的置信水平，也无须多次运行算法来研究不同置信水平下的解。通过考量可用的解及其利润相关风险，可以针对问题实例，更充分地判断应采用哪些解。此外，我们还引入了一种过滤方法，该方法会定期应用于适应度评估过程中。它只保留中间种群中具有有效 α 区间的解，使下一代产生的新子代解能够在这些解的基础上进一步改进。实验研究证明了这些方法的性能。

致谢

本研究得到了澳大利亚研究理事会（ARC）通过项目编号 FT200100536 提供的资助，以及南澳大利亚州政府通过研究联盟“通过精益加工释放复杂资源”提供的资助。本研究还使用了阿德莱德大学 Phoenix 高性能计算服务提供的超级计算资源。

参考文献

Yuma Abe、Masaki Ogura、Hiroyuki Tsuji、Amane Miura 和 Shuichi Adachi。2020年。卫星通信系统的资源与网络管理：机会约束方法。IFAC 53（2020年），3304–3309。第2期

Hirad Assimi、Oscar Harper、Yue Xie、Aneta Neumann 和 Frank Neumann。2020年。基于尾界目标的动态机会约束背包问题进化双目标优化。载于 ECAI 2020，第325卷。IOS Press，307–314。C.C. Coello、D.A. Van Veldhuizen 和 G.B. Lamont。2013年。求解多目标问题的进化算法。Springer US。Kalyanmoy Deb。2001年。使用进化算法进行多目标优化。John Wiley & Sons。Kalyanmoy Deb、Samir Agrawal、Amrit Pratap 和 T. Meyarivan。2002年。一种快速且具有精英策略的多目标遗传算法：NSGA-II。IEEE进化计算汇刊 6，2（2002年），182–197。Benjamin Doerr。2020年。随机化优化启发式算法分析的概率工具。Springer，第1章，1–87。Benjamin Doerr、Carola Doerr、Aneta Neumann、Frank Neumann 和 Andrew M. Sutton。2020年。机会约束次模函数优化。载于第三十四届AAAI人工智能会议，AAAI 2020。AAAI出版社，1460–1467。Benjamin Doerr 和 Frank Neumann。2021年。离散优化进化算法理论近期进展综述。ACM进化学习与优化汇刊 1，4，文章16（2021年10月），43页。耿新波和谢磊。2019年。具有概率保证的电力系统数据驱动决策：机会约束优化理论与应用。《控制年度评论》47（2019年），341–363。O. Giel。2003年。一种简单多目标进化算法的期望运行时间。载于2003年进化计算大会，2003年。CEC '03，第3卷。1918–1925，第3卷。何方国和邵桂明。2009年。一种不确定优化问题的进化算法。载于2009年国际信息工程与计算机科学会议。IEEE，1–4。Konstantinos Kepaptsoglou、Grigorios Fountas 和 Matthew G. Karlaftis。2015年。封闭海域天气对集装箱船航线规划的影响。《交通运输研究C辑：新兴技术》55（2015年），139–155。刘波、张清福、Francisco V. Fernández 和 Georges G. E. Gielen。2013年。一种高效的机会约束双目标随机优化进化算法。IEEE进化计算汇刊 17，6（2013年），786–796。Daniel H. Loughlin 和 S. Ranji Ranjithan。1999年。机会约束遗传算法。载于 GECCO '99。Morgan Kaufmann出版社，369–376。Kazuyuki Masutomi、Yuichi Nagata、和 Isao Ono。2013年。一种黑箱机会约束函数优化的进化算法。《高级计算智能与智能信息学杂志》17，2（2013年），272–282。Mehrnaz Mohtasham、Hossein Mirzaei-Nasirabad 和 Behrooz Alizadeh。2021年。不确定条件下露天矿卡车—铲土机分配优化：一种机会约束目标规划方法。《采矿技术》130（2021年），81–100。Aneta Neumann 和 Frank Neumann。2020年。使用进化多目标算法优化单调机会约束次模函数。载于《自然界并行问题求解》—PPSN XVI—第16届国际会议，PPSN 2020，论文集，第一部分（《计算机科学讲义》，第12269卷）。Springer，404–417。Aneta Neumann、Yue Xie 和 Frank Neumann。2022年。限制随机利润背包问题中不确定性影响的进化算法。载于 PPSN XVII。Springer，Cham，294–307。Frank Neumann、Mojgan Pourhassan 和 Vahid Roostapour。2020年。动态随机环境下进化算法的分析。Springer，第7章，323–357。Frank Neumann 和 Andrew M. Sutton。2019年。用于机会约束背包问题的 (1 + 1) 进化算法运行时间分析。载于 FOGA '19（FOGA '19）。ACM，147–153。Frank Neumann 和 Carsten Witt。2022年。正态分布随机变量机会约束优化问题的单目标与多目标进化算法运行时间分析。载于 IJCAI-22。4800–4806。彭申。2019年。机会约束问题及其应用。学位论文。巴黎-萨克雷大学（COmUE）；西安交通大学。石峰、严先坤和 Frank Neumann。2022年。机会约束最大完工时间调度问题的简单进化算法运行时间分析。载于 PPSN XVII。Springer，526–541。

Hemant Kumar Singh 和 Jürgen Branke。2022年。使用进化计算识别随机非支配解。载于 PPSN (2)（《计算机科学讲义》，第 13399 卷）。Springer，193–206。Yue Xie、Oscar Harper、Hirad Assimi、Aneta Neumann 和 Frank Neumann。2019。机会约束背包问题的进化算法。载于 GECCO '19。ACM，338–346。Yue Xie、Aneta Neumann 和 Frank Neumann。2020。用于机会约束背包问题的特定单目标与多目标进化算法。载于 GECCO '20。ACM，271–279。Yue Xie、Aneta Neumann 和 Frank Neumann。2021a。求解具有机会约束的复杂交互式库存混配问题的启发式策略。载于 GECCO '21。ACM，1079–1087。Yue Xie、Aneta Neumann、Frank Neumann 和 Andrew M. Sutton。2021b。RLS 和 (1+1) EA 求解具有相关均匀权重的机会约束背包问题的运行时间分析。载于 GECCO '21。ACM，1187–1194。