doi:10.3772/j.issn.1002-0470.2018.04.001

基于多目标贪心策略的增益最大化团队构建算法①

宋永浩②[*,**] 史 骁[*,**] 胡 斌[*] 金 岩[*] 杜翠兰[***] 井雅琪[***] 赵晓芳[*]

([*]中国科学院计算技术研究所 北京 100190)

([**]中国科学院大学计算机与控制学院 北京 101408)

([***]国家计算机网络应急技术处理协调中心 北京 100029)

摘 要 研究了满足一定约束条件的协同作业团队的构建。针对传统构建忽略了团队成员在协作过程中个体技能可增加这一因素,提出了团队构建的统一优化目标函数问题,并在综合考虑协同作业任务所需技能集合覆盖约束和团队成员之间交流代价最小化约束的基础上,引入了团队成员增益最大化约束。针对该多目标优化问题,提出了3种基于贪心策略的启发式团队构建算法,即基于最小集合覆盖贪心策略的团队构建算法——贪心集覆盖算法(GSCA)、基于团队增益最大化贪心策略的团队构建算法——贪婪团队增益算法(GTGA)和基于多路径(MR)贪心策略的团队构建算法——MRGTGA。大量实验证明,GSCA 较适用于交流代价极高的远程协作环境,MRGTGA 较适用于对算法运行效率要求不高、但对整体增益最大化要求极高的场景,GTGA 构建的团队整体增益值接近精确解(其值达到暴力枚举算法的96.70%),同时该算法运行效率极高(其计算时间接近 GSCA)。

关键词 团队构建,团队增益,交流代价,多目标优化,贪心策略

0 引 言

随着社会的快速发展,各行业项目开展所需技能知识日趋复杂和多样化,如何构建恰当的协同团队则成为行业项目能否成功的关键[1]。近年,自动化团队构建问题作为基础科学问题,被广泛应用到大型公司的人力资源组织[2]、教育领域协同学习团队构建[3]、工程项目实施团队组建[4]、团体运动项目成员选择[5]等领域。然而团队构建问题受不同领域应用场景的影响,其存在多样化的约束条件,例如集合覆盖约束、团队成员间交流代价最小化约束等,并且此类团队构建问题已被证明为 NP 完全问题[3,6,7]。本研究在引入团队成员增益最大化约束的基础上,给出了3种基于贪心思想的增益最大化团队构建方法,并对其进行了详细论述和实验验证。

1 相关研究

团队构建问题是指:给定一项需要若干特定技能才能完成的复杂任务,一个能够参与的人员集合,其中每个人员掌握了若干技能,目标是构建满足一定约束条件的团队以实施给定的任务[6-9,10]。

团队构建问题的基本约束条件是构建团队的整体技能集合能够覆盖任务需要的技能集合,但是基本约束并不能满足实际的应用场景需求。在此基础上,当前相关研究主要针对不同的应用场景,增加贴近现实状况的约束条件。现有的团队构建约束大致可以分为4类。第一类考虑团队之间的交流代价。交流代价是影响团队协作效率的关键因素。文献[6]利用团队成员的社交关系结构,构建了两种直观的交流代价函数:子图直径交流代价函数和最

① 国家自然科学青年基金(No. 61202413)资助项目。

② 男,1989 年生,博士生;研究方向:机器学习与人工智能;联系人,E-mail: songyonghao@ict.ac.cn

(收稿日期:2017-10-28)

小生成树交流代价函数。基于不同的交流代价函数,产生了两种启发式交流代价最小化团队构建算法:RarestFirst 和 EnhancedSteiner。文献[8]在团队交流代价的基础上,引入人力资源成本约束,构建了最小化交流代价与人力资源成本的双目标团队构建任务,并通过迭代替换的启发式算法进行求解。第二类考虑团队中是否存在领导者。团队领导者的选取对完成任务存在不同程度的积极影响。文献[7]讨论了是否存在团队领导者约束条件下的团队构建问题,并基于上述约束提出了两种不同的团队交流代价:团队成员两两距离总和以及团队成员与领导者距离总和,形式化证明了上述问题是 NP 难度的,并给出近似比为 2 的启发式算法。文献[1]基于人员的知识背景与协同关系提出了团队构建模型,通过专业技能、交流技巧、团队建设、领导力和协作能力 5 个方面选择团队领导者。文献[9]分析了之前研究工作中选择团队领导者带来的巨大计算开销,通过降低候选者的搜索空间,提出了两种高效的团队领导者发现算法:BCPruning 与 SSPruning。第三类引入团队成员的能力约束或序列任务的负载均衡约束。任务协同存在并行化特征,需要解决成员同时参与过多任务导致过载的问题。文献[10]综合考虑任务技能覆盖约束和团队成员的能力约束,提出了 MinAggrSol 启发式算法来求解上述双目标优化问题。文献[11]引入在线序列任务加载场景,着重强调序列任务分配的负载均衡性,指出所有的团队构建问题都是 NP 难度的,并提出一种高效的启发式算法求解。文献[12]指出在线社交媒体环境下用户有吸引其他用户参与的能力,该文献定义了“参与度团队构建”问题,并基于博弈理论方法解决上述问题。文献[13]在同样测试标准下对目前主流团队构建算法进行了对比和分析,并开源了统一的团队构建系统。文献[14]在序列任务的场景下,实现了能够收益最大化的团队构建方案。

现有的团队构建研究普遍忽略了团队增益这一重要因素。实际场景下,团队成员在协同作业过程中能够相互学习对方技能,实现自我价值提升[3,15],从而实现团队增益的提高。本文在满足任务技能覆盖约束和团队成员交流代价最小化约束的基础上,引入团队成员整体增益最大化约束,给出了增益最大化的团队构建问题的形式化定义,并从定量计算的角度提出了 3 种基于贪心策略的团队构建算法,其中以基于整体增益最大化贪心策略的团队构建算法——贪婪团队增益算法(greedy team gain algorithm, GTGA)综合效果最优,该算法在为团队增加候选成员时,引入使团队增益最大的候选成员,既要满足任务技能覆盖约束,同时最大化团队整体增益值,并通过交流代价约束限制团队规模。

2 符号说明及问题定义

2.1 符号说明

下面介绍团队构建问题中涉及到的符号表示及其含义,如表 1 所示。

表 1 问题定义符号说明

符号	含义
E	候选人员集合
N	候选人员数量
S	总体技能集合
M	总体技能数量
T	任务所需要的技能集合
X	构建团队的成员集合
$S(e)$	候选人员 e 拥有的技能集合
$S(X)$	团队 X 中所有成员拥有技能集合的并集

为不失一般性,本文假设存在 N 位候选人员,表示为 $E=\{e_1,e_2,\cdots,e_N\}$,存在 M 项特定技能 $S=\{s_1,s_2,\cdots,s_M\}$。某位候选人员 e_i 拥有的技能表示为 $S(e_i)$,则 $S(e_i)\subseteq S$;拥有技能 s_k 的候选人员集合表示为 $E(s_k)$,则 $E(s_k)\subseteq E$。给定一个需要若干技能才能完成的复杂任务,表示为 $T=\{s_i,\cdots,s_j\}\subseteq S$,如果构建的团队 X 是任务 T 的一个可行团队,则必须要求团队 X 所有成员拥有技能集合的并集能够覆盖完成任务 T 所需的技能集合,即 $T\subseteq\cup_{e_i\in X}S(e_i)$。

2.2 问题定义

本文中涉及到的团队构建问题,是指给定一个任务 T 和候选人员集合 E,目标是为任务 T 构建一

个“最优”协同作业团队,其中“最优”涉及三方面的约束条件:

(1) 技能覆盖约束:要求团队中所有成员拥有技能集合的并集能够覆盖任务 T 所需的技能;

(2) 交流代价最小化约束:团队成员数量的增加会导致团队协作的交流成本增加,需要团队交流代价最小化;

(3) 团队增益最大化约束:团队成员在协作过程中相互学习,能够获得一定的增益值,每位成员的增益值之和作为团队增益,并期望团队增益最大化。

基于上述团队构建问题的描述,给出如下形式化定义。

定义 1:团队中单个成员的增益函数。给定构建的协同团队 $X=\{e_1,e_2,\cdots,e_n\}$,其中 $n<<N$,则团队 X 拥有的技能集合为 $S(X)=\cup_{e_i\in X}S(e_i)$,定义团队中单个成员 e_i 的增益函数为:$g(e_i)=|S(X)\backslash S(e_i)|$。

定义 1 中采用团队拥有的技能集合中单个成员未掌握的技能数量作为该成员的增益值,因此团队增益表示为 $g(X)=\sum_{e_i\in X}g(e_i)$。

定义 2:团队交流代价函数。给定构建的协同团队 $X=\{e_1,e_2,\cdots,e_n\}$,$|X|$ 表示团队成员数量,假设团队中任意两个成员之间的交流代价为 1,则团队整体的交流代价表示为:$c(X)=\frac{|X|(|X|-1)}{2}$。

定义 2 中交流代价仅同团队成员的个数相关。

本文研究的团队构建问题的目标是在满足技能集合覆盖的基础上,构建团队增益最大化与交流代价最小化的协同团队,该问题是双目标优化问题,此类问题的一类经典求解思路是将两个优化目标进行有机组合[8]。本文通过平衡参数 λ 将团队增益与交流代价定义为统一的优化目标函数。

定义 3:统一优化目标函数。给定构建的协同团队 $X=\{e_1,e_2,\cdots,e_n\}$,团队增益与交流代价之间的平衡参数 λ,定义构建团队 X 的统一优化目标函数为

$$ComGainCost(X)=\sum_{e_i\in X}g(e_i)-\lambda\times\frac{|X|(|X|-1)}{2}$$

根据上述定义,给出本文所研究的团队构建问题的形式化定义。

问题 1:统一优化目标函数最大化的团队构建问题。给定任务 $T=\{s_i,\cdots,s_j\}$,候选人员集合 $E=\{e_1,e_2,\cdots,e_N\}$,其中每位候选人员拥有一个技能集合 $e_i=\{s_k,\cdots,s_l\}$,以及团队增益与交流代价的平衡参数 λ。本文研究的团队构建问题是:构建协同团队 X,X 拥有的技能能够覆盖任务 T 所需技能,即 $T\subseteq S(X)$,并且使得定义的统一优化目标函数 $ComGainCost(X)$ 最大化(定义 3)。针对上述团队构建的优化问题定义如下:

$$Maximize\ ComGainCost(X)=\sum_{e_i\in X}g(e_i)-\lambda\times\frac{z(z-1)}{2}$$

$$\text{Subject to: } z=\sum_{i=1}^{|E|}Q_i$$

$$A\cdot Q\geqslant I^{|T|}$$

$$Q\in\{0,1\}^{|E|}$$

$$A\in\{0,1\}^{|T|\times|E|}$$

为了实现后续论述的简洁性,本研究将问题 1 表示为 CGC-Team-formation。

CGC-Team-formation 问题可以通过归约为经典集合覆盖问题(set cover problem, SCP),来说明该问题是 NP 完全的。CGC-Team-formation 问题在不考虑增益最大化约束的条件下,即为 SCP 问题,可以自然地得出 CGC-Team-formation 属于 NP 完全问题。

2.3 示例描述

为了更加清晰地展示上述问题 1 中定义的团队构建问题,本节给出软件系统开发领域中具体场景下团队构建问题示例描述。假设存在一个 web 系统开发任务 $T=\{$HTML, CSS, JS, Java, Spring, Tomcat, Mysql, Linux$\}$,以及一组候选工程师,其中每位工程师都拥有掌握的技能集合,如下表所示。

表 2 候选工程师及其拥有技能

工程师	拥有技能
甲	HTML, JS, CSS, Tomcat
乙	HTML, JS, Java, Spring, Tomcat, Mysql, Linux
丙	Mysql, Linux
丁	Java, Spring, python

由甲和乙工程师构成的团队拥有技能集合

{HTML, CSS, JS, Java, Spring, Tomcat, Mysql, Linux},该团队可以覆盖开发任务 T 所需技能,因此团队{甲,乙}是上述团队构建问题的一个可行解。由定义 1 可以计算出工程师甲在该团队中获得增益值为 8 −4 =4,工程师乙在该团队中获得增益值为 8 −7 =1,因此团队{甲,乙}增益值为 4 +1 =5;由于该团队只有 2 名成员,由定义 2 可以计算出该团队的交流代价为 2(2 −1)/2 =1;由定义 3 中给出的统一优化目标函数,当平衡参数 $\lambda = 1$ 时,团队{甲,乙}的整体增益值为团队增益减去团队成员交流代价,即 5 −1 =4。同样计算出该团队构建问题的所有可行解如表 3 所示。

表 3 上述团队构建问题所有可行解

团队成员	团队增益	团队成员交流代价	团队整体增益
甲,乙	4 +1 =5	2(2 −1)/2 =1	5 −1 =4
甲,乙,丙	4 +1 +6 =11	3(3 −1)/2 =3	11 −3 =8
甲,乙,丁	5 +2 +6 =13	3(3 −1)/2 =3	13 −3 =10
甲,丙,丁	5 +7 +6 =18	3(3 −1)/2 =3	18 −3 =15
甲,乙,丙,丁	5 +2 +7 +6 =20	4(4 −1)/2 =6	20 −6 =14

由表 3 可以得出,团队{甲,丙,丁}的整体增益值最大,为上述 web 系统开发任务 T 的最佳协同团队。

3 协同团队构建算法分析

由于 CGC-Team-formation 是 NP 完全问题,采用贪心策略的方法实现问题的启发式求解是处理 NP 完全问题的有效方法。通常,贪心策略未必能提供该类问题的最优解,但可以通过不同贪心策略对最优解进行逼近。为提供策略效果的对比,本节首先介绍了利用暴力枚举算法获取最优解的方法,随后介绍了采用不同贪心策略实现的协同团队构建算法。

3.1 暴力枚举算法

假设存在任务 $T = \{s_i, \cdots, s_j\}$,候选人员集合 $E = \{e_1, e_2, \cdots, e_N\}$,候选人员数量 $N = |E|$。首先枚举所有候选人员存在的组合,每一种组合构成任务 T 的协同团队,则共有 $\sum_{k=1}^{N} C_N^k = 2^N - 1$ 种不同的协同团队,然后遍历所有团队判断是否满足技能覆盖约束,并计算满足技能覆盖约束团队的增益值,选取增益值最大的团队作为任务 T 的最佳协同作业团队。由于该算法需要枚举所有形式的组合团队以及选出增益值最大的团队,故该算法时间复杂度为 $O(2^N N)$。暴力枚举算法能够求得问题的精确解,并且可以在小规模数据场景下评估启发式算法的效果,该算法为暴力枚举(brute-force enumeration, BE)算法。

3.2 基于最小集合覆盖的贪心策略

该策略借鉴最小集合覆盖问题的一种近似贪心算法[16],在团队构建过程中满足技能覆盖约束条件,并且使得团队规模接近最小,降低成员交流代价,但是由于团队规模较小,团队成员整体增益值也会较小。该算法叫做贪心集覆盖算法(greedy set cover algorithm, GSCA),算法伪代码如算法 1 所示。

算法 1: formTeam _ GSCA (T,E,λ)

输入: T: 任务所需的技能集合; E: 候选人员集合; λ: 团队增益值与交流代价的平衡参数

输出:构建的协同团队 *bestTeam*

```
1 bestTeam ←{∅, ∅, − ∞ }//{X:团队成员,C:团队拥有
                          //技能集合,teamGain:团队增益值}
2 U ← T
3 while U ≠∅
4     选择使得 | S(e) ∩ U | 最大的候选成员 e, e ∈ E
5     bestTeam. X 增加成员 e
6     bestTeam. C 增加技能集合 S(e)
7     U←U − S(e)
8 bestTeam. teamGain←ComGainCost(X, λ)   //定义 3
9 return bestTeam
```

针对上述贪心集覆盖算法(GSCA)的伪代码,对其核心算法原理进行进一步分析。在每个阶段,集合 U 包含余下未被覆盖的元素构成的集合;集合 *bestTeam* 包含团队成员集合 X,团队拥有的技能集合 C,以及当前阶段该团队的增益值 *teamGain*。第4行是贪心决策步骤,即选出一个候选成员 e,使它能覆盖尽可能多的未被覆盖的任务所需技能。在 e 被选出后更新 *bestTeam*,即向团队成员集合 X 中增加成员 e(第5行),第6行将成员 e 拥有的技能集合 $S(e)$ 添加到团队技能集合 C,第7行将 e 拥有技能从 U 中去掉。当循环终止时,*bestTeam* 即为满足任务 T 技能集合覆盖约束的协同团队,第8行计算该团队的增益值。

算法 GSCA 是以 $|T|$ 和 $|E|$ 的多项式时间运行。因为第3~7行间循环体的迭代次数至多为 $\min(|T|, |E|)$,而且可以将循环体实现以时间 $O(|E||T|)$ 运行,故存在一个运行时间为 $O(|E||T|\min(|T|, |E|))$ 的实现。

3.3 基于团队增益最大化的贪心策略

该策略在满足构建团队拥有的技能集合覆盖任务所需技能的约束条件下,最大化构建团队的整体增益值。团队整体增益值采用定义3中的统一优化目标函数表示。该策略的核心是在构建团队的过程中如何寻找下一个最佳的(能够使得团队增益值最大)候选成员,通过不断向团队中增加能够使得团队增益增大的候选成员,实现构建团队的增益最大化,值得注意的是随着团队规模的增大,团队的交流成本也会随之增加,为了使得最终构建的团队整体增益值最大化,需要实现团队增益值与团队成员交流代价之间的平衡。该算法表示为贪婪团队增益算法(GTGA),算法伪代码如算法2所示。

算法 2: formTeam _ GTGA (T,E,λ)

输入: T: 任务所需的技能集合; E: 候选人员集合; λ: 团队增益值与交流代价的平衡参数

输出:构建的协同团队 *bestTeam*

```
1  bestTeam ← {∅,∅, -∞} //{X:团队成员,C:团队拥有
                        // 能技集合,teamGain:团队
                        增益值}
2  U ← T
3  while U ≠∅
4      e ← findBestCandidate(bestTeam,E,λ)
5      bestTeam. X 增加成员 e
6      bestTeam. C 增加技能集合 S(e)
7      U←U - S(e)
8  bestTeam. teamGain←ComGainCost(X, λ)   //定义3
9  while true
10     e ←findBestCandidate(bestTeam,E,λ)
11     if e = null then break
12     else
13         bestTeam. X 增加成员 e
14         bestTeam. C 增加技能集合 S(e)
15 bestTeam. teamGain ← ComGainCost(X, λ)   //定义3
16 return bestTeam
```

针对上述算法 GTGA 的伪代码,对其核心算法原理进行进一步分析。在每个阶段,集合 U 与集合 *bestTeam* 的含义以及包含的元素与算法 GSCA 相同。第4行是贪心决策步骤,即选出一个最佳候选成员 e,使它加入当前团队,能对团队增益值的贡献最大,具体原理见后续方法1的详细分析。在 e 被选出后,需要同步更新集合 *bestTeam* 和集合 U(第5~7行)。当第一个循环体(第3~7行)迭代结束后,能够保证当前构建的团队满足任务 T 所需技能集合的覆盖约束,但并不能保证该团队是增益值最大的团队,第二个循环体(第9~14行)继续向当前团队中增加能够使得团队增益值增大的候选人,直至 findBestCandidate 方法无法找到新的候选人迭代终止(第11行)。最后第16行计算该团队的增益值。

算法2中关键方法 *findBestCandidate*(*team*,E,λ) 的伪代码如方法1所示。

方法 1: *findBestCandidate*(*team*,E,λ)

输入: *team*: 当前构建的团队; E: 候选人员集合; λ: 团队增益值与交流代价的平衡参数

输出:最佳候选成员 *candidate*

```
1  tempTeam ← team
2  tempTeamGain ← - ∞
3  for e ∈ E do
4      向 tempTeam 中增加成员 e   //同步更新 team. X,
                                  // team. C, teamGain
5      if tempTeam. teamGain > tempTeamGain then
```

```
6        candidate ← e
7        tempTeamGain ← tempTeam. teamGain
8    从 tempTeam 中移除成员 e   //同步更新 team. X,
                                // team. C, teamGain
9  return candidate
```

方法 1 的作用是查找能够使得当前构建团队的增益值增加最大的候选人员。具体过程是将候选人员集合中候选人逐个加入到当前团队(第 3 行),并计算加入某候选人之后当前团队的增益值(第 4 行),由 *candidate* 保存能够使得当前团队增益值增加最大的候选人(第 5 ~ 7 行)。当遍历完候选人员集合,则选出最佳候选人员,需要注意的是当团队规模不断增加,团队交流代价也会呈多项式增加,导致新增一名团队成员,反而可能会使得团队增益值下降,此种情况下方法 1 会返回空值。

算法 GTGA 是以$|T|$和$|E|$的多项式时间运行。因为第 3 ~ 7 行间循环体的迭代次数至多为 $\min(|T|, |E|)$,第 9 ~ 14 行间循环体的迭代次数至多为$|E| - \min(|T|, |E|)$;方法 1 中第 3 ~ 8 行间的循环体迭代次数为$|E|$,而且可以将循环体实现以时间$O(|T|)$运行,故方法 1 的时间复杂度为$O(|E||T|)$;由于算法 GTGA 两个循环体的主要操作都是方法 1,并且两个循环体的迭代次数总和为$|E|$,故 GTGA 存在一个运行时间为$O(|E|^2|T|)$的实现。

3.4 基于任务新手引入与团队增益最大化的双路径贪心策略

在课题研究过程中发现将任务新手加入协同团队,能够提升团队整体增益值,因为在协同过程中对任务新手的技能提升更显著。任务新手是指掌握的技能数量很少,或者很少掌握当前任务所需的技能。从实际生产环境中,也存在老师傅带新徒弟的情况,进一步说明在构建的协同团队中引入少量新手能够提升团队整体增益值。但是算法 GTGA 在贪心过程中只关注使得团队增益最大化的候选成员,并且当前路径的最优选择并不是全局最优选择。因此本文提出了基于任务新手引入与团队增益最大化的双路径贪心策略,该策略在候选成员选择过程中保持两条搜索路径。实验结果显示,该算法进一步逼近了暴力枚举求出的精确解。该算法表示为多路径贪婪团队增益算法 MRGTGA(multi-route GTGA, MRGTGA),算法伪代码如算法 3 所示。

算法 3: formTeam _ MRGTGA (T,E,λ)

输入: T: 任务所需的技能集合; E: 候选人员集合; λ: 团队增益值与交流代价的平衡参数

输出:构建的协同团队 *bestTeam*

```
1  candidateTeamSet ← ∅
2  随机构造一个种子团队,添加到 candidateTeamSet
3  while true
4    tempCandidateTeamSet ← ∅
5    for team ∈ candidateTeamSet do
6        tempTeam ← team
7        bestCandidate ← findBestCandidate(team,E, λ )
8        tempTeam. X 增加成员 bestCandidate
9        tempTeam. C 增加技能集合 S(bestCandidate)
10         向 tempCandidateTeamSet 中添加 tempTeam
11         possibleCandidate ← findPossiCandidate(team,E)
12         team. X 增加成员 possibleCandidate
13         team. C 增加技能集合 S(possibleCandidate)
14         向 tempCandidateTeamSet 中添加 team
15   candidateTeamSet ← tempCandidateTeamSet
16   if possibleCandidate = null then break
17 找出 candidateTeamSet 中最优团队 bestTeam
18 return bestTeam
```

针对上述算法 MRGTGA 的伪代码,对其核心算法原理进行进一步分析。相比于算法 GTGA,MRGTGA 采用双路径贪心策略,第 7 行采用的贪心决策步骤与算法 GTGA 相同,即选出一个能使团队增益值增加最大的候选成员 *bestCandidate*,第 11 行是第二个贪心决策步骤,即选出一个当前团队中不存在的,并且掌握技能数量最少的候选人 *possibleCandidate*。将此两个候选人分别添加到当前团队中,则产生出两个临时团队,第 10 和 14 行将此两个临时团队添加到 *tempCandidateTeamSet* 集合中,第 5 ~ 14行间的循环体遍历保存的所有临时团队,并对每个团队按照两种贪心决策步骤添加候选人员。当无法找到新的 *possibleCandidate*(第 16 行)时,第 3 ~ 16 行循环体终止,第 17 行从候选团队集合中找出增益值最大的团队,作为任务 T 的最优执行团队。

MRGTGA 的时间复杂度是$O(|E|^2 \log_2^{|E|} |T|)$。第一层循环体(第 3 ~ 16 行)的迭代次数至多为$|E|$,

第二层循环体(第 5 ~ 14 行)的迭代次数计算相对复杂,由于该循环体内采用双路径贪心策略,搜索过程实际构成一棵满二叉树,树的深度为 $\log_2^{|E|+1}$, *candidateTeamSet* 中临时团队的数量符合以 2 为底的等比数列,由求和公式得 *candidateTeamSet* 中累计存在 $2\log_2^{|E|+1}-1$ 个临时团队,故第二层循环体的迭代次数为 $O(\log_2^{|E|})$,而且可以将第二层循环体实现以时间 $O(|E||T|)$ 运行;因此算法 MRGTGA 存在一个运行时间为 $O(|E|^2\log_2^{|E|}|T|)$ 的实现。

4 实验仿真

4.1 仿真数据设置

为了验证本文算法之间的差异性以及算法有效性,本研究在不同数据规模、多种数据分布条件下进行了大量的实验,下面对仿真数据的设置进行详细介绍。每组实验需要包含一项依赖若干技能才能完成的任务,一个候选人集合及每位候选人的技能集合。针对每组实验本文首先确定相应的元数据记录,包括实验序号、候选人数量、任务所需技能数量、所有候选人拥有的技能总数量以及每组实验的重复次数。根据实际应用环境,本文中实验结果涉及到的数据规模设置为:候选人数量 N 从 8 逐渐增长到 128,总体技能数量 M 从 16 逐渐增长到 128,为保证任务所需技能数量在合理范围内,将其设置为 $(\log_{2.5}^{M})^2$,每组实验重复次数为 10 次。

根据仿真的元数据记录生成候选人员的技能集合以及任务所需技能集合。针对每一条元数据记录,根据任务所需技能数量以及实验重复次数,随机生成 10 项任务技能集合。根据元数据记录中候选人数量以及总体技能数量,生成每位候选人拥有的技能集合,并且一组实验中候选人拥有的技能数量按照均匀分布、高斯分布与帕累托分布进行生成,其中均匀分布表示本组实验中候选人之间技能分布比较均匀,高斯分布表示候选人拥有的技能数量更接近客观世界,帕累托分布表示有少数候选人拥有大量技能。总体技能数量为 M, 本实验中高斯分布采用的均值为 $M/4$, 标准差为50.0,帕累托分布中随机变量可能取到的最小值参数为0.45,指数参数为 $2.0/\log_2(M)$。

4.2 实验结果及分析

本文实验平台采用曙光 I620-C20 机架服务器,处理器为 Intel Xeon E5-2630 v3,主频为 3.2GHz,运行内存为 32GB,操作系统为 CentOS7。数据的仿真以及算法实现的编程语言采用 Java,其中不同数据分布的产生采用 Apache 开源的数学工具库 commons-math3-3.6.1.jar[17]。

由于现有团队构建相关研究基于不同的约束场景,目前没有算法能在有效时间内解决本研究提出的团队构建问题。本实验部分主要对提出的暴力枚举(BE)算法、GSCA、GTGA 以及 MRGTGA 团队构建算法进行一个详细的对比评测。主要讨论随着数据规模的增长算法的有效性对比分析,不同数据分布条件下构建团队的特点,算法在计算时间方面的对比,以及平衡参数 λ 对团队规模的影响。

4.2.1 算法效果对比分析

为了验证提出算法的有效性,在小数据规模上,分别运行暴力枚举算法(BE)、基于最小集合覆盖的贪心算法(GSCA)、基于团队增益最大化的贪心算法(GTGA)以及基于任务新手引入与团队增益最大化的双路径贪心算法(MRGTGA)。其中 BE 算法能够获得问题的精确解。图 1 和图 2 设置候选人员拥有的技能数量符合高斯分布,平衡参数 λ 设置为 1。其中图 1 固定候选人员数量为 24,总体技能数量由 16 增加至 64;图 2 设置总体技能数量为 32,候选人员数量由 8 增加至 24。由图 1 可以看出在小数据规模上,算法 GTGA 和 MRGTGA 构建的协同团队的整体增益值已经非常接近 BE 算法求解的精确解,经过统计计算,算法 MRGTGA 求解的增益值与 BE 算法求解的增益值比值的平均值为98.38%,算法 GTGA 与 BE 算法求解的增益值比值的平均值为 96.70%,算法 GSCA 获取团队的平均增益值为 BE 算法的9.29%。图 2 显示随着候选人数量的增加,BE 算法运行时间呈指数型增长,而其他 3 个贪心算法运行时间增长缓慢。图 2 中算法 GTGA 的运行时间接近 GSCA,而算法 MRGTGA 的运行时间约为 GSCA 算法的 3 倍。因此 BE 算法由于其指数级的时间复杂度,只适用于小数据规模场景下;MRGTGA

较适用于对算法运行效率要求不高、但对整体增益最大化要求极高的场景。

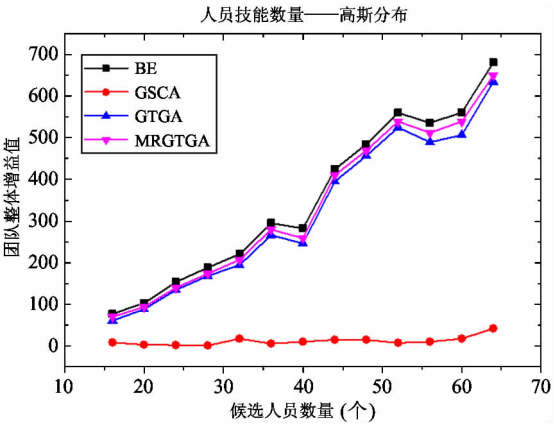

图 1　小数据规模条件下算法有效性对比

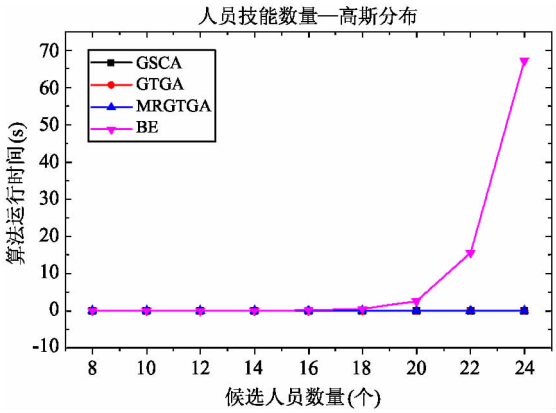

图 2　小数据规模条件下算法运行效率对比

图 3 和图 4 主要用来说明随着数据规模的增长，并且在不同的数据分布条件下，算法 GTGA 与 MRGTGA 构建的协同团队仍能获得近似最优化的团队整体增益值。图 3 实验中设置总体技能数量为 64，候选人员数量由 16 增加至 128，图 4 中设置候

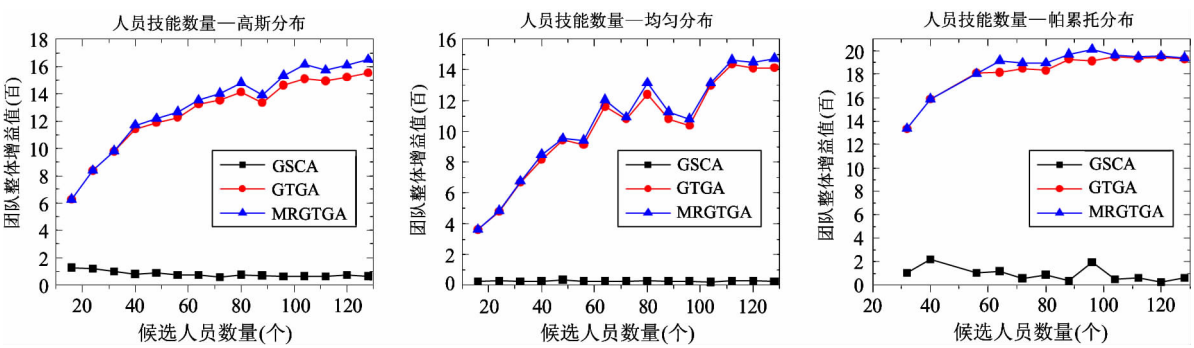

图 3　固定总体技能数量条件下不同贪心策略算法效果对比

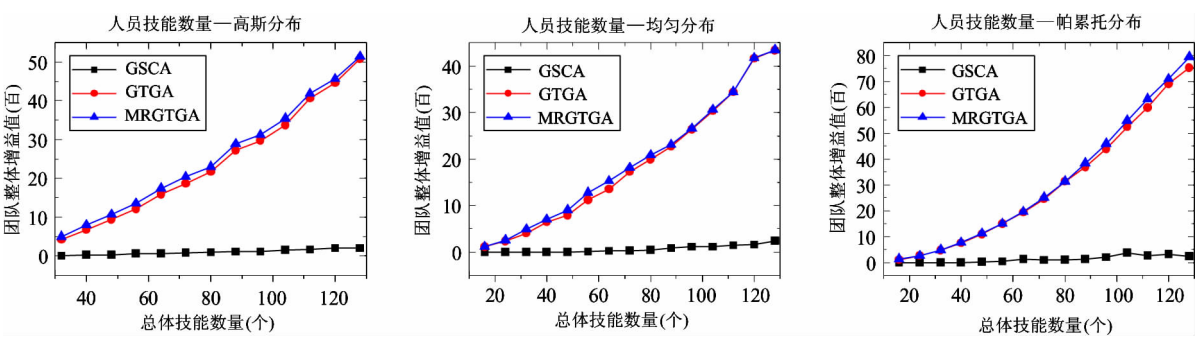

图 4　固定候选人员数量条件下不同贪心策略算法效果对比

选人员数量为 128，总体技能数量由 16 增加至 128，并且两组实验中候选人员拥有的技能数量分别符合高斯分布、均匀分布和帕累托分布。由图 3 和图 4 可以看出即使在不同数据分布条件下，算法 GTGA 构建团队的整体增益值与算法 MRGTGA 非常接近，并且团队增益值远高于算法 GSCA 构建团队的增益

值。由于算法 GSCA 的贪心策略是最小化团队成员数量,构建团队的整体增益值变化缓慢,在总体技能数量不变的情况下,候选人员数量的增加甚至会导致算法 GSCA 输出的增益值缓慢下降;反之,在候选人员数量不变的情况下,总体技能数量的增加使算法 GSCA 输出的增益值缓慢上升。由图 3 和图 4 对比可以看出,在候选人数量不变的情况下,算法 GTGA 和 MRGTGA 获得的团队整体增益值增长比较线性,并且有加快的趋势;另外在总体技能数量不变的条件下,随着候选人员数量的增加,算法 GTGA 和 MRGTGA 获得的团队整体增益值增长逐渐变缓,这是由于即使候选人员数量在增加,实际存在的最优团队的增益值也会被固定的总体技能数量所限制。

在固定总体技能数量,候选人数量不断增加的场景下(图 3),经过统计计算,算法 GSCA 计算的团队增益值是 MRGTGA 算法的8.08%,算法 GTGA 计算的团队增益值是 MRGTGA 算法的96.23%。在固定候选人数量,不断增加总体技能数量场景下(图 4),经过统计计算,算法 GSCA 计算的团队中增益值是算法 MRGTGA 的5.37%,算法 GTGA 计算的团队增益值是算法 MRGTGA 的98.51%。综合上述分析,多种数据分布条件下,算法 GSCA 构建团队的整体增益值较小,而算法 GTGA 在不同的数据分布下,获得的团队增益值都拥有较高的精确度。

4.2.2 数据分布对团队构建的影响

现实世界中不同行业领域中的技术人员所拥有的技能数量会服从不同的数据分布,例如普通技术行业从业人员拥有的技能数量服从高斯分布或者均匀分布,高精尖科学领域专家的技能数量服从帕累托分布,只有少数专家掌握大量技能。本研究还探索了候选人员拥有的技能数量服从不同数据分布对团队增益的影响,图 5 显示,算法 GTGA 在帕累托分布条件下,能够获得最大的团队整体增益值,在均匀分布条件下,获得的增益值最小。这是由于在帕累托分布条件下,人员拥有的技能数量离散程度最大,从而构建的团队中成员拥有的技能数量差异化较大,拥有技能数量少的成员能获得更大的增益值;而在均匀分布条件下,人员拥有的技能数量比较均匀,相互之间学习获得的增益较小。技能数量服从高斯分布的情况,恰好介于两者之间。

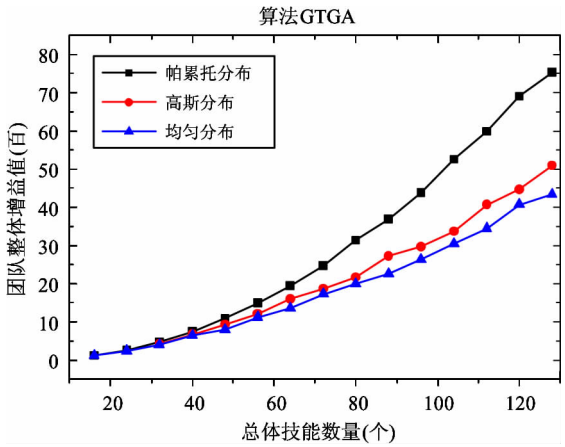

图 5 候选人技能数量分布对团队增益的影响

4.2.3 算法计算时间对比

图 6 显示随着候选人员数量的增加,算法 GSCA 与 GTGA 的运行时间增长缓慢,然而算法 MRGTGA 运行时间增长显著,并且从算法效果对比分析小节中可知算法 MRGTGA 获得的最大团队增益值并不明显优于 GTGA 算法。因此综合评价算法 GTGA 效果最优,GTGA 构建的团队整体增益值接近精确解,其值达到暴力枚举算法的96.70%,同时该算法运行效率极高,其计算时间接近 GSCA。图 7 显示算法 GTGA 在帕累托分布条件下,能够更快地搜索到最优团队,在均匀分布条件下搜索到最优团队需要消耗更多时间,这是由于*findBestCandidate*()

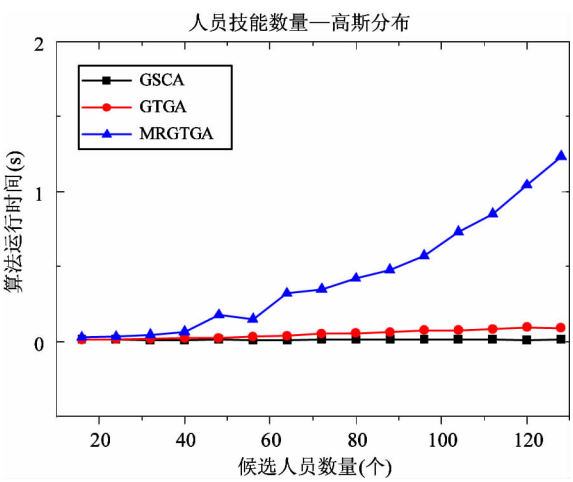

图 6 算法间运行时间对比

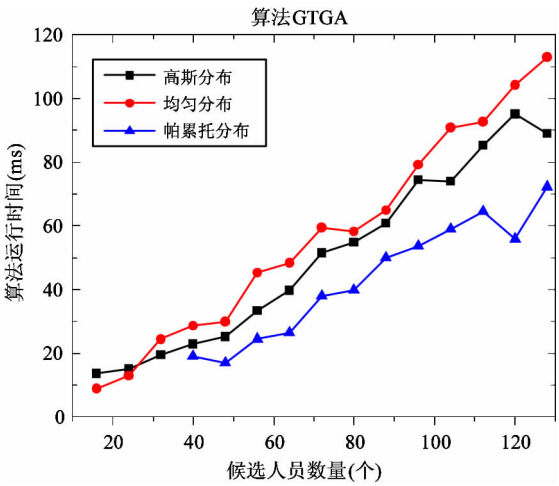

图 7　不同数据分布条件下算法运行时间对比

在候选人员拥有的技能数量离散化程度更高的情况下,更能快速搜索到最佳的团队成员,从而减少后续操作。

4.2.4　平衡参数对团队规模的影响

为了平衡团队增益最大化与团队交流代价最小化的双重优化目标,我们引入了平衡参数 λ。当 $\lambda = 0$ 时,本研究定义的团队构建问题的优化目标变成团队增益最大化的单一目标,该种情况下,向团队中引入新成员至少不会使得团队整体增益降低,因此贪心算法会不断向当前团队中加入新成员,直到加入所有候选人员,图 8 显示当 $\lambda = 0$ 时,算法 GSCA 与 MRGTGA 最终构建的团队成员数量达到了 128,即将所有候选人员构建成一个协同团队。当 $\lambda \to +\infty$ 时,主要的优化目标是团队成员交流代价最小化。本研

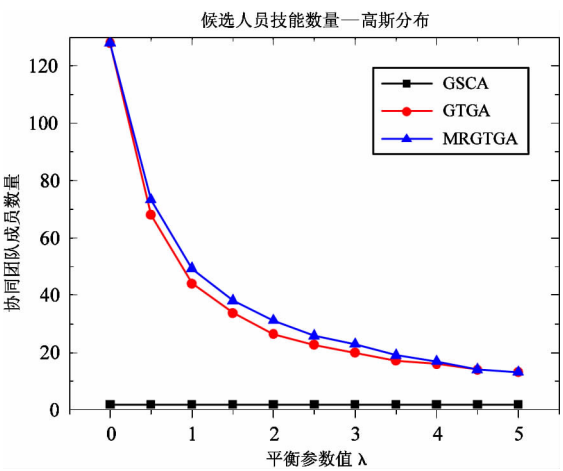

图 8　平衡参数对团队规模影响

究定义的团队构建问题近似于经典最小集合覆盖问题,由图 8 表明随着平衡参数 λ 的不断增大,算法 GSCA 与 MRGTGA 构建的团队成员数量不断下降,并逐步逼近最小集合覆盖问题的贪心算法 GSCA 的最优解。此外由图 8 可以看出,算法 GSCA 构建团队的成员数量较小,相应带来的交流代价也极小,从而适用于交流代价极高的远程协作环境。

4.3　算法特性总结

仿真实验部分主要在数据规模增长的条件下,对本文提出的三种算法的有效性、运行时间以及构建团队规模等维度进行分析,从而突出不同算法的适用场景。实验结论总结如表 4 所示。

表 4　算法特性总结

算法名称	时间复杂度	算法有效性
BE	$O(2^{\lvert E\rvert} \lvert E\rvert)$	100%
GSCA	$O(\lvert E\rvert \min(\lvert T\rvert, \lvert E\rvert))$	9.29%
GTGA	$O(\lvert E\rvert^2 \lvert T\rvert)$	96.7%
MRGTGA	$O(\lvert E\rvert^2 \log_2^{\lvert E\rvert} \lvert T\rvert)$	98.38%

表 4 中 $|E|$ 为候选人员数量,$|T|$ 表示任务所需技能数量,其中 BE 算法能够获得本文定义团队构建问题的精确解,其他三种启发式算法的有效性,是在平均意义上与 BE 算法精确解的比值。综合以上实验分析,BE 算法只适用于小数据规模的团队构建问题;GSCA 较适用于交流代价极高的远程协作环境;MRGTGA 较适用于对算法运行效率要求不高、但对整体增益最大化要求极高的场景;GTGA 构建的团队整体增益值接近精确解,同时该算法运行效率极高。

5　结　论

本文在传统团队构建问题的基础上,引入团队成员增益最大化的优化目标,形式化定义了统一优化目标函数最大化的团队构建问题,通过将该问题归约为经典集合覆盖问题,说明了该问题的 NP 完全性。然后通过对问题本质的深入分析,提出了多种基于贪心策略的启发式团队构建算法。实验结果表明,算法 MRGTGA 求解的团队整体增益值为暴力

枚举算法的98.38%,算法 GTGA 效果为暴力枚举算法的 96.7%,并且随着数据规模的增长,算法 GTGA 运行时间增加缓慢,可以适用于候选人员数量和领域技能数量大规模的场景。另外,本文对候选人员技能数量服从的不同分布以及平衡参数的设置对团队构建效果的影响进行探讨。实验表明候选人员的技能数量分布越集中,越容易搜索到团队构建的最优解,并且构建的团队整体增益值也会越大;平衡参数的增大,会导致构建团队所需成员数量的减少,并且引起团队增益值的下降。本文定义的团队构建问题,没有依赖具体的应用领域,提出的算法具备一定的通用性。

目前对于团队交流代价的计算,假设任意两个成员间的交流代价为 1,没有考虑团队成员间的社交关系,在进一步研究工作中将通过社交关系来定义成员间交流代价。

参考文献

[1] Wi H, Oh S, Mun J, et al. A team formation model based on knowledge and collaboration[J]. *Expert Systems with Applications*, 2012, 40(1):9121-9134

[2] Stranad D, Guid N. A fuzzy-genetic decision support system for project team formation[J]. *Journal of Applied Soft Computing*, 2010, 10(4):1178-1187

[3] Agrawal R, Golshan R, Terzi E. Grouping students in educational settings[C]. In: Proceedings of the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, New York, USA, 2014. 1017-1026

[4] Yang Y, Chen W J, Yang J. Team formation in business process context[C]. In: Proceedings of the 21th IEEE International Conference on Computer Supported Cooperative Work in Design, Wellington, New Zealand, 2017. 73-78

[5] Budak G, Kara Í, Íç Y T, et al. New mathematical models for team formation of sports clubs before the match [J]. *Central European Journal of Operations Research*, 2017(1):1-17

[6] Lappas T, Liu K, Terzi E. Finding a team of experts in social networks[C]. In: Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Paris, France, 2009. 467-475

[7] Kargar M, An A. Discovering top-k teams of experts with/without a leader in social networks[C]. In: Proceedings of the 20th International Conference on Information and Knowledge Management (CIKM), Scotland, UK, 2011. 985-994

[8] Kargar M, An A, Zihayat M. Efficient bi-objective team formation in social networks[C]. In: Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases, Bristol, UK, 2012. 483-498

[9] Juang M C, Huang C C, Huang J L. Efficient algorithms for team formation with a leader in social networks[J]. *Journal of Supercomputing*, 2013, 66(2):721-737

[10] Datta S, Majumder A, Naidu K. Capacitated team formation problem on social networks[C]. In: Proceedings of the 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Beijing, China, 2012. 1005-1013

[11] Anagnostopoulos A, Becchetti L, Castillo C. Online team formation in social networks[C]. In: Proceedings of the 21th International Conference on World Wide Web, Lyon, France, 2012. 839-848

[12] Nikolaev A, Gore S, Govindaraju V. Engagement capacity and engaging team formation for reach maximization of online social media platforms[C]. In: Proceedings of the 22th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, San Francisco, USA, 2016

[13] Wang X Y, Zhao Z, Ng W. USTF: A unified system of team formation[J]. *Journal of IEEE transactions on big data*, 2016, 2(1):70-84

[14] Tang S J. Profit-driven team grouping in social networks [C]. In: Proceedings of the 31th AAAI Conference on Artificial Intelligence, San Francisco, USA, 2017. 45-51

[15] Garrett J, Gopalakrishna S. Sales team formation: the right team member helps performance[J]. *Journal of Industrial Marketing Management*, 2017. Doi:10.1016/j.indmarman.2017.06.007

[16] 殷建平,徐云,王刚等译. 算法导论[M]. 第三版. 北京:机械工业出版社,2014. 658

[17] Apache Organization. The apache commons mathematics library[EB/OL]. http://commons.apache.org/proper/commons-math:Apache, 2017

Team formation algorithms with team gain maximization based on multi-objective greedy strategy

Song Yonghao[*,**], Shi Xiao[*,**], Hu Bin[*], Jin Yan[*], Du Cuilan[***], Jing Yaqi[***], Zhao Xiaofang[*]

([*] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

([**] School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 101408)

([***] National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029)

Abstract

The formation of the collaboration teams meeting certain constraints is studied. Considering that traditional team formation ignores that team members can enhance their skills during the collaboration, the problem of unified objective function optimization for team formation is proposed, and based on the comprehensive consideration of cover constraints for skill sets required for collaborative tasks and minimization constraints of exchange costs among team members, a team gain maximization constraint is also introduced. To solve the multi-objective optimization problem, three heuristic team formation algorithms based on greedy strategies are proposed to maximize the unified objective function. They are the team formation algorithm based on minimum set cover: greedy strategy, the greedy set cover algorithm (GSCA); team gain maximization greedy strategy; the greedy team gain algorithm (GTGA) and multi-route GTGA (MRGTGA), respectively. The extensive experimental results indicate that GSCA is more appropriate for the (remote) collaboration scenario with extremely high communication cost, MRGTGA is more appropriate for the collaboration scenario where entirety team gain maximization is required, no matter how complexity of the algorithm, and the entirety gain of the team formed by GTGA is approximately close to optimum (The value is 96.70% of the brute-force enumerate algorithm), meanwhile GTGA has the extremely high algorithm efficiency (The computation time is close to GSCA).

Key words: team formation, team gain, communication cost, multi-objective optimization, greedy strategy