

基于拥挤距离的动态粒子群多目标优化算法

魏 武, 郭 燕

(华南理工大学 自动化科学与工程学院, 广东 广州 510640)

摘 要 :提出了一种改进的基于拥挤距离的动态粒子群多目标优化算法。为提高粒子的全局搜索能力,提出了新的动态变化惯性权重和加速因子的方法。引进了拥挤距离排序方法维护外部精英集和更新全局最优值。为保持非劣解的多样性,采用了小概率变异机制,并根据种群的大小选择不同的变异概率。最后,把算法应用到5个典型的多目标测试函数并与其他算法进行比较。实验结果表明,该算法所得的 Pareto 解集有很好的收敛性和多样性。

关键词 :多目标优化; 拥挤距离; 粒子群; 惯性权重; 外部精英集; 非劣解

中图分类号 :TP301.6 文献标识码 :A 文章编号 :1000-7024 (2011) 04-1422-04

Dynamic particle swarm algorithm for multi-objective optimization based on crowding distance

WEI Wu, GUO Yan

(College of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China)

Abstract : An improved dynamic particle swarm algorithm for multi-objective optimization based on crowding distance is proposed. To explore the global space more efficiently, the inertia weight and acceleration coefficients are dynamically changed. Meanwhile, the crowding distance sorting is used to maintain the external elitist archive and select the global social leaders. To keep the diversity of the non-dominated solutions, the mutation operator mechanism is adopted, and the probability of mutation is selected according to the size of the population. At last, the algorithm is applied to five classical test functions and compared to other algorithms. It is shown from the results that the Pareto solution obtained from this strategy has a good convergence and diversity.

Key words : multi-objective optimization; crowding distance; particle swarm; inertia weight; external elitist archive; non-dominated solutions

0 引 言

在科学研究和工程实践中,经常会碰到多目标优化问题,像航空系统、电力系统、卫星导航系统、机器人控制系统等,因此成功解决多目标优化问题是一个非常具有实际意义和科研价值的课题。通常,多目标问题与单目标问题有很大的区别,它并不是普通单一的最值问题,需要考虑多个目标值之间的冲突和协调,它的解是一组解的集合,通常称为非劣解集或 Pareto 解集。

粒子群多目标优化(multi-objective particle swarm optimization, MOPSO)算法的主要优点是它的收敛速度快,有利于多目标优化算法的实际工程应用。Coello Coello 等^[1]提出的引入外部档案保存非劣解,以指导其他粒子的飞行;Liang 等^[2]提出了改进的全面学习粒子群算法;Huang 等^[3]利用该算法处理多目标优化问题;此外还有学者也提出了一些多目标混合粒子群算法^[4-5]。R.A.Santana 等^[6]提出了基于拥挤距离和赌盘选择法

的多目标粒子群算法(MOPSO-CDR)。从以上的文献来看, MOPSO 算法现有的主要缺点是: 全局搜索能力弱而容易陷入局部极值,降低了算法的寻优能力; 算法的随机性大,容易出现波动。

本文采用文献[7]中拥挤距离排序的思想,为了提高粒子群算法的全局搜索能力和保持非劣解的多样性,提出了一种新的通过动态改变惯性权重和加速参数的方法来更新粒子的速度的多目标粒子群算法,并采用小概率变异机制。通过对5个经典多目标测试函数进行30次的随机测试结果表明,该算法所得的 Pareto 解集具有很好的全局收敛性和分散性,随机性误差较低。

1 理论基础

1.1 粒子群算法介绍

粒子群优化算法^[8](particle swarm optimization, PSO)是一种群体智能算法,它是由美国心理学家 Kennedy 和电器工程师

收稿日期:2010-04-21; 修订日期:2010-06-21。

基金项目:国家自然科学基金重点项目(60736024); 中央高校基本科研业务费基金项目(2009ZM192)。

作者简介:魏武(1970-),男,湖南益阳人,博士,教授,研究领域为机器人控制、模式识别与人工智能; 郭燕(1986-),女,湖南娄底人,硕士研究生,研究方向为机器人运动规划与智能优化。E-mail: guo.yan@mail.scut.edu.cn

Eberhart 在 1995 年首先提出来的。PSO 算法是将每个个体看作是搜索空间的一个没有体积和质量的粒子,在搜索空间中以一定的速度飞行,并且根据对个体和集体的飞行经验的综合分析来动态调整飞行速度。标准 PSO 中,粒子在搜索空间中的速度和位置是根据下面的公式确定

$$v(t+1) = wv(t) + c_1 \cdot r(t) \cdot (Pbest - x(t)) + c_2 \cdot r(t) \cdot (Gbest - x(t)) \quad (1)$$

$$x(t+1) = x(t) + v(t+1) \quad (2)$$

式中: w ——惯性权重系数,能够使粒子保持运动惯性,使其有扩展搜索的趋势,有能力探索新的区域; $Pbest$ 和 $Gbest$ 分别为第 t 代时粒子的自身最好位置和全局最好位置; c_1 和 c_2 为加速常数,代表将每个粒子推向 $Pbest$ 和 $Gbest$ 位置的统计加速项的权重; $r(t)$ 为 $[0, 1]$ 上均匀分布的随机数; $v(t)$ 与 $x(t)$ 分别是粒子在第 t 代时的速度与位置,在更新的过程中,粒子的位置坐标必须限制在所考虑的范围之内。

1.2 多目标优化介绍

多目标优化问题一般定义如下:

定义 1 一般 MOP 由 n 个决策变量、 M 个目标函数和 K 种约束条件组成,用公式表达为

$$\begin{aligned} \min y = f(x) &= [f_1(x), f_2(x), \dots, f_M(x)] \\ \text{s.t.} \\ g(x) &= 0, i = 1, 2, \dots, p \\ h(x) &= 0, i = 1, 2, \dots, q \end{aligned} \quad (3)$$

式中: $x = (x_1, x_2, \dots, x_n)$ D 为决策向量; $y = f(x) = [f_1(x), f_2(x), \dots, f_M(x)]$ Y 为目标向量; D 为决策向量形成的决策空间; Y 表示目标向量形成的目标空间。多目标优化问题中,经常会用到以下几个基本概念:

定义 2

(1) Pareto 支配: 解 x^0 支配 x^1 , 当且仅当

$$\begin{aligned} f_i(x^0) &\leq f_i(x^1), i = \{1, 2, \dots, M\} \\ f_i(x^0) &< f_i(x^1), i = \{1, 2, \dots, M\} \end{aligned}$$

(2) Pareto 最优: 如果解 x^0 是 Pareto 最优的当且仅当不存在 $x^1: x^1$ 支配 x^0 。

(3) Pareto 最优集: 所有 Pareto 最优解的集合 $P_s = \{x^0 | \text{不存在 } x^1 \text{ 支配 } x^0\}$ 。

(4) Pareto 最优前端均衡面或 Pareto 前端: 所有 Pareto 最优解对应的目标函数值所形成的区域 $P_F: P_F = \{f(x) = (f_1(x), f_2(x), \dots, f_M(x)) | x \in P_s\}$ 。

2 基于拥挤距离的动态多目标粒子群算法

2.1 拥挤距离的计算和排序

拥挤距离是用来估计一个解的周围其他解的密集程度。对于每个目标函数,先对外部精英解集中的解根据目标函数的大小进行排序,然后计算每个解临近的两个粒子所构成的立方体的平均边长,最终的结果就是这个解的拥挤距离(边界解的拥挤距离为无穷大)。计算所得拥挤距离越小,则所得的解越密集,其多样性越小,反之说明所得解越稀疏,多样性越大。最后根据计算所得拥挤距离的大小对粒子重新进行降序排列。

2.2 惯性权重系数和加速因子的维护

粒子的搜索过程主要包括全局搜索和局部搜索。在算法

的运行中,如果全局搜索比较强,则很容易忽略局部的最优值,而如果局部搜索过于强,很容易使粒子陷入局部最优。因此在调整参数的过程中,要尽量避免出现两种极端状况,需要协调以得到更好的 Pareto 解集。

在式(1)中,各个参数的变化均会引起粒子搜索范围的变化。惯性权重系数 w 能够保持原有的运动趋势, w 大能够促使粒子探索新的空间范围,有利于全局搜索,反之,则有利于局部搜索。因此,本文在此基础上,提出了一种新的动态调整 w 值的方法:随着迭代次数的增加,每一代的惯性权重 $w(t)$ 的最大值根据式(4)从 0.9 到 0.5 线性减小,式中, t 为当前迭代次数, $maxgen$ 为最大迭代次数。同时,为保持粒子的多样性,本文对每个粒子的惯性权重值 $w_i(t)$ 的最大值按照式(5)分配,最后在区间 $[w_i(t) - 0.3, w_i(t)]$ 随机选择一个作为粒子的惯性权重系数。

$$w(t) = -0.4 \frac{t}{maxgen} + 0.9 \quad (4)$$

$$w_i(t) = -0.3 \frac{i}{popsize} + w(t) \quad (5)$$

加速因子 c_1 和 c_2 用来平衡个体和群体的认知能力。 c_1 和 c_2 有较低的值,则允许粒子在被拉回之前可以在目标区外徘徊,而较高的值则导致粒子突然冲向或者越过目标区域。为平衡粒子在搜索空间中的探测和开发能力,保证粒子的全局寻优能力,在本文中为使得到最后群体认知部分占主导地位,通过实验设计 c_2 保持 2.0 不变,而 c_1 在每次迭代中则按照式(6)在 1.5 到 0.8 之间进行线性递减

$$c_1(t) = -0.7 \frac{t}{maxgen} + 1.5 \quad (6)$$

通过上述配置参数之后,对速度的更新按照式(7)进行整体调整, k 表示一个 0.7 至 1.0 的随机数。

$$v(t+1) = k [wv(t) + c_1 \cdot r(t) \cdot (Pbest - x(t)) + c_2 \cdot r(t) \cdot (Gbest - x(t))] \quad (7)$$

2.3 搜索空间内变量的维护和粒子的变异

为了保证解的可行性并使搜索逼近某一些可能的区域,一旦变量值超过了一定的边界(即粒子不在可行域内),就需要对粒子进行调整。本文通过限定粒子的飞行边界,当粒子飞出了规定的边界,则令粒子的当前值为边界值(最大或者最小值),同时对该粒子的飞行速度取反,使下一代粒子逆着本代飞行方向在搜索空间内飞行。在本文中,根据粒子群的大小选择粒子的变异概率,以更好的抑制粒子群算法的收敛速度。按照公式(8)计算,变异概率最小值设为 0.01,最大值设为 0.5。

$$pMut = \frac{20}{popsize} (popsize - 50) \quad (8)$$

2.4 外部精英集的更新

外部精英集是用来保存粒子在搜索过程中产生的非劣解,在这里我们称其为精英解。当迭代次数为 t 时,如果粒子群所产生的非劣解支配了精英集中的成员,则删除被支配的解,并以外部的非支配解代替。如果外部精英集 M 的个体数没有达到最大容量值 m ,则将粒子群搜索产生的非支配解复制到外部精英集,形成新的外部精英集 $M1$ 。首先判断新的外部精英集中是否有重复的个体(目标值相同的个体),如果有,则只保留一个个体,其他重复个体全部删除。此时新的外部精英集 $M1$ 中解的个数 n 如果没有超过最大容量值,则进行粒

子的下一代的搜索;如果 n 超过最大容量值,重新计算 $M1$ 中各个解的拥挤距离并通过拥挤距离降序排列。最后,删除 $M1$ 中后面 $n-m$ 个距离拥挤的个体,得到另一个精英解集 $M2$,此时 $M2$ 中的非劣解的个数为 m 。

2.5 全局最优值的选取

通过对拥挤距离的计算和排序后,全局最优值将在拥挤距离比较大的个体群中随机选出。其计算方法如式(9), $nondom$ 为非劣解个数, $A[j]$ 为精英解集中拥挤距离较大中的随机的一个。此时随机选出的 $Gbest$ 是一个处于 Pareto 前端中分散区域的个体。

$$\begin{aligned} i &= 0.1(nondom-1) \\ j &= RandomInt(0,i) \\ Gbest &= A[j] \end{aligned} \quad (9)$$

2.6 算法的实现步骤

(1)对粒子群 P 中各个粒子进行初始化,包括初始化粒子的位置,并计算出初始位置的目标函数值,初始化粒子的飞行速度为 0,外部精英集的最大容量值 $m=200$,保存粒子的局部最优值 $Pbest$,将初始粒子中非劣精英解置入到外部精英集;

(2)当外部精英集中解得个数超过 3 个的时候,计算每个非劣解的拥挤距离并且根据拥挤距离大小进行降序排列;

(3)对式(1)中各个参数按照 2.2 所述的方法进行调整,按照式(9)选取全局最优值 $Gbest$,然后根据式(7)和式(2)分别更新粒子的速度和位置。按照 2.3 所述的维护变量的搜索空间范围,适当调整粒子的速度;

(4)粒子在速度更新之后,在满足条件 $t < maxgen \cdot pMut$ 的情况下,对粒子进行小概率变异。

(5)分别计算每个粒子目标值,然后更新粒子的局部最优值,同时按照 2.4 更新外部精英集,迭代次数 $t=t+1$ 。

当 $t \leq maxgen$ 时重复步骤(2),否则,结束算法,输出外部精英集,获得 Pareto 近似解集。

3 实验测试及结果分析

3.1 多目标优化测试的性能指标

在多目标优化的研究中,通常需要用一些性能指标来评价该算法是否实现了要求达到的目标。本文中主要用到以下两个性能指标来评价算法^[6]。

(1)非劣解在目标空间上的分布范围(分散性) SP ,其定义为

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (10)$$

式中 $d_i = \min_j (\sum_{k=1}^M |f_k^i - f_k^j|)$, $i, j = 1, 2, \dots, n, i \neq j$, $\bar{d} = \frac{1}{n} \sum_{i=1}^n d_i$, N ——算法求得的非劣解的个数, d_i ——第 i 个非劣解的目标值到其他所有非劣解目标值的最小距离, M ——目标函数的个数, \bar{d} ——最短距离的期望值。 SP 越小,表示算法所得的 Pareto 解集前端分布越均匀。

(2)最大散布范围 D ,其定义为

$$D = \sqrt{\sum_{k=1}^M (\max f_k^i(x) - \min f_k^j(x))^2} \quad (11)$$

它表示测量空间中 Pareto 解集中的两个极值解的距离,通过对 D 的测量,可以检测所得非劣解的散布范围。 D 越大,

表示所获得解得范围越广。

3.2 数值实验对比分析

为评价本文粒子群算法的性能,选用目标测试函数 ZDT1~ZDT6^[9] 作为测试实例,选择函数的决策变量维数 $m=10$ 。针对不同的测试函数,算法的主要参数设置如下:各个函数的种群大小 $popsiz = 100$,最大迭代代数 $maxgen$ 根据测试函数的性质,ZDT1 和 ZDT2 设置为 200,ZDT3 设置为 300,而 ZDT4 和 ZDT6 由于存在局部极值点,因此其计算的最大迭代次数 $maxgen=2000$ 。惯性权重 w 和加速常数 c_1 和 c_2 均按照 2.2 所述设置,精英档案大小 $m=200$,每个测试函数计算 30 次。

测试函数 ZDT1~ZDT3 的表达式如下

$$\begin{aligned} \text{ZDT1} \quad & \begin{cases} \min f_1(x) = x_1 \\ \min f_2(x) = g(1 - \sqrt{x_1/g}) \\ g = 1 + 9(\sum_{i=2}^n x_i/(n-1)) \end{cases} \end{aligned} \quad (12)$$

$$x_i \in [0, 1] \quad i = 1, 2, \dots, n$$

$$\begin{aligned} \text{ZDT2} \quad & \begin{cases} \min f_1(x) = x_1 \\ \min f_2(x) = g[1 - (x_1/g)^2] \\ g = 1 + 9(\sum_{i=2}^n x_i/(n-1)) \end{cases} \end{aligned} \quad (13)$$

$$x_i \in [0, 1] \quad i = 1, 2, \dots, n$$

$$\begin{aligned} \text{ZDT3} \quad & \begin{cases} \min f_1(x) = x_1 \\ \min f_2(x) = g(1 - \sqrt{x_1/g} - x_1 \sin(10\pi x_1)/g) \\ g = 1 + 9(\sum_{i=2}^n x_i/(n-1)) \end{cases} \end{aligned} \quad (14)$$

$$x_i \in [0, 1] \quad i = 1, 2, \dots, n$$

在 30 次的测试中,ZDT1 和 ZDT2 迭代 200 次运行时间均维持在 10s 左右,ZDT3 迭代 300 次的运行时间仅为 4s 左右。通过分别对这 3 个函数的测试,所得的 Pareto 前端如图 1 所示。从图 1 中可以看出本算法所得的解都为最优解,而且各个测试函数的 Pareto 前端分布均匀。

本文算法获得的 3 个函数的测试性能指标结果如表 1 所示,与文献[10]中的混合粒子群优化算法相比,本文所得 ZDT1 和 ZDT3 的平均 SP 值均要小,证明本文算法所得非劣解有更好的收敛性。与文献[6]中提及的 MOPSO, m -DNPSO, MOPSO-CDLS, CSS-MOPSO, MOPSO-CDR 这 5 种算法相比较,本文算法的迭代次数要远远小于上述算法的,所得 ZDT1 的各项性能的综合仅略差于 CSS-MOPSO,ZDT2 与 ZDT3 的结果与 MOPSO-CDR 相当。而且从表格中所得标准差相比,可以得知本文所得的 30 次结果 SP 和 D 的波动范围比较小,所得的值基本上能够一致,表明每次运算的随机误差较小。

测试函数 ZDT4 和 ZDT6 的表达式分别如下

$$\begin{aligned} \text{ZDT4} \quad & \begin{cases} \min f_1(x) = x_1 \\ \min f_2(x) = g(1 - \sqrt{x_1/g}) \\ g = 1 + 10(n-1) + \sum_{i=2}^n (x_i^2 - 10\cos(4\pi x_i)) \end{cases} \end{aligned} \quad (15)$$

$$\text{除 } x_1 \in [0, 1] \text{ 外, 其余 } x_i \in [-5, 5] \quad i = 2, 3, \dots, n.$$

$$\begin{aligned} \text{ZDT6} \quad & \begin{cases} \min f_1(x) = 1 - \exp(-4x_1) \sin^6(6\pi x_1) \\ \min f_2(x) = g[1 - (x_1/g)^2] \\ g = 1 + 9(\sum_{i=2}^n x_i/(n-1))^{0.25} \end{cases} \end{aligned} \quad (16)$$

$$x_i \in [0, 1] \quad i = 1, 2, \dots, n$$

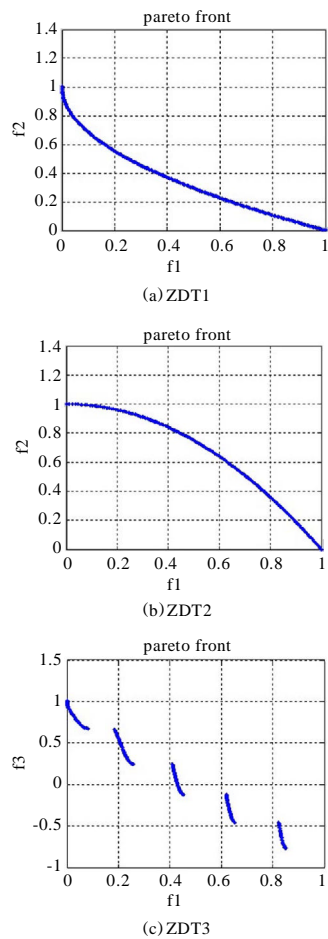


图 1 测试函数 ZDT1~ZDT3 的 Pareto 解的分布

表 1 测试函数 ZDT1~ZDT3 的 SP 值和 D 值

函数	SP				D			
	Max	Min	Ave	Std	Max	Min	Ave	Std
ZDT1	0.0036	0.0032	0.0034	3.1E-5	1.4163	1.4148	1.4152	7.7E-5
ZDT2	0.0032	0.0035	0.0033	2.9E-5	1.4146	1.4142	1.4143	1.2E-5
ZDT3	0.0037	0.0030	0.0034	5.2E-5	1.9777	1.9715	1.9741	7.1E-4

ZDT4 由于存在局部极值,是一个很难优化的函数,在大多数文献中很少提及对它的测试。本文经实验测试所得的 Pareto 前端如图 2 (a) 所示,上方曲线为本算法所得 Pareto 前端,下方为真实 Pareto 前端,从图 2 中可以得知本文所得的 Pareto 解集中没有一个是最优解,但是与文献[11]比较,在粒子数相同,迭代次数相同,本文所得的近似 Pareto 前端更加逼近于真实 Pareto 前端,而且解的分布也更均匀,所得 SP = 0.0037。随着迭代次数的增加,逼近程度会有所增加,不过当迭代次数达到一定程度的时候(10000 次以上),始终无法再逼近真实解,有待于进一步解决。

ZDT6 所得到的非劣解分布如图 2 (b) 所示。对于测试函数的 30 次实验测试中,其运行时间均在 20s 左右。每次所得的 200 个非劣解中平均有 2 个非有效解,算法稍微有些波动。在 30 组数据中,SP 的最大值为 0.1730,最小值为 0.0026,所得平均值为 0.0382,其标准方差为 0.009。对比文献[6]中的各个算法所得的 SP 值,可知本文所得的 Pareto 前端分布是最均匀

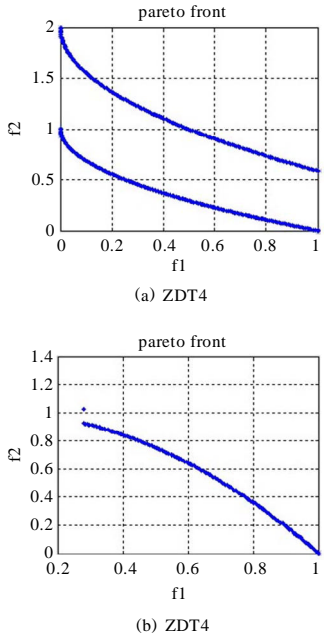


图 2 测试函数 ZDT4 和 ZDT6 的 Pareto 解的分布

的。如图 2(b)中的 Pareto 前端的分布情况来看,通过对比文献[12]中的实验结果,本算法的收敛性和分散性均更优。

4 结束语

在对现有的 MOPSO 研究的基础基于拥挤距离更新外部精英集和选择全局最优值的思想,为提高粒子的全局搜索能力,提出了一种新的动态改变权重系数的方法。本文的粒子群算法,可以权衡粒子的全局搜索能力和局部搜索能力,使粒子搜索所得的非劣解正确率高,能够更好的逼近真实 Pareto 前端。并且使粒子具有多样性,保证了粒子的 Pareto 前端分布的均匀性。通过对实验结果数据的分析,与其他算法进行比较,证实本文的算法能够减小计算的随机误差,运行时间短,同时对于一些复杂问题(如 ZDT4, ZDT6)的解决,有了相当的改进,明显提高了粒子的全局收敛性。

参考文献:

[1] Coello Coello C A,Pulido G T,Lechuga M S.Handling multiple objectives with particle swarm optimization[C].IEEE Transactions on Evolutionary Computation,2004,8(3):256-279.

[2] Liang J J,Qin A K,Suganthan P N,et al.Evaluation of comprehensive learning particle swarm optimizer[C].Lectures Notes in Computer Science.Berlin :Springer,2004:230-235.

[3] Huang V L,Suganthan P N,Liang J J.Comprehensive learning particle swarm optimizer for solving multi-objective optimization problems [J]. International Journal of Intelligent Systems, 2006,21:209-226.

[4] Santana-Quintero L V,Ramirez N,Coello Coello C A.A multi-objective particle swarm optimizer hybrid with scatter search [C].Lecture Notes in Computer Science,MICAI 2006: Advances in Artificial Intelligence,2006:294-304. (下转第 1452 页)

态,得到正向搜索动作序列为库所 ActionSequences 中的托肯所示,而后向搜索动作序列由动作序列库所 ActionSequenceReverse 中托肯所示。

从图2中可以看出,其中库所 ActionSequences 中存储的正向搜索动作序列为:

open(d34),enter(303),go(da3),enter(alley)

而库所 ActionSequenceReverse 中存储的后向搜索的动作序列为

enter(301),go(da1),enterReverse(alley),go_reverse(da3)

对其进行逆排后得到

go_reverse(da3),enterReverse(alley),go(da1),enter(301)

将逆排后的动作序列其与前面正向搜索的动作序列连接,去掉正向搜索后部分和后向搜索动作序列前部分最大重复的子串(此重复子串是由于从正向和反向进行搜索时达到共同的状态时所进行的重复动作),得到最终的动作序列为

open(d34),enter(303),go(da3),enter(alley),go(da1),enter(301)

此动作序列对应了从初始状态到目标状态的一个成功的规划,左边的time中标注了动作执行时间为8个单位,此时间为机器人实际执行此动作序列所需要的执行时间,接下来,机器人可以进行重新规划,如指定目标位置所在的门为“d12”,重新开始进行规划,从而可以从中选择一条执行时间最短的最优路径。

4 结束语

本文提出了一种基于情景演算的前提条件公理、后继状态公理以及初始状态公理来进行规划的形式化方法——BSCRPN网系统,与形式化系统情景演算相比,BSCRPN网系统有以下优点:

(1)BSCRPN解决了框架问题,弥补了情景演算不能解决框架问题的缺陷;

(2)增加了对动作时间的表示,使得此规划系统能满足对

实时要求高的场合需要;

(3)由于情景演算采用单向搜索机制实现对目标的查找而生成动作序列,而BSCRPN网则采用双向搜索策略来搜索目标,从而具有更高的执行效率。

本文的下一步工作为将双向搜索策略用于多机器人规划实例中,改善多机器人的规划效率。

参考文献:

- [1] Kurt Jensen,Lars Michael Kristensen,Lisa Wells.Coloured Petri nets and CPN tools for modelling and validation of concurrent systems[J].Software Tools for Technology Transfer,2007,9(3-4): 213-254.
- [2] 于振华,贾俊秀,蔡远利.基于Petri网的多Agent系统设计与实现[J].电子科技大学学报,2007,36(3):517-520.
- [3] Palamara P F,Ziparo V A,Iocchi L,et al.A robotic soccer passing task using Petri net plans[C].Proc of 7th Int Conf on Autonomous Agents and Multiagent Systems,2008:1711-1712.
- [4] Vittorio Amos Ziparo,Luca Iocchi.Petri net plans[C].Fourth International Workshop on Modelling of Objects,Components,and Agents,2006:267-290.
- [5] 杜卓敏,何炎祥.基于有色网的多Agent计划建模[J].计算机科学,2006,33(6):154-157.
- [6] John McCarthy,Patrick J Hayes.Some philosophical problems from the standpoint of artificial intelligence[M].Edinburgh:Machine Intelligence 4,Edinburgh University Press,1969:463-502.
- [7] Raymond Reiter.Knowledge in action:Logical foundations for specifying and implementing dynamical systems[C].Cambridge, MA:MIT Press,2001.
- [8] CPN Tools.A computer tool for colored Petri nets[EB/OL]. <http://www.daimi.au.dk/CPNTools/>,2008.

(上接第1425页)

- [5] Liu D,Tan K C,Goh C K,et al. A multi-objective evolutionary algorithms based on particle swarm optimizations[J]. IEEE Transactions on Systems, Man and Cybernetics, Part B, 2007,37 (1): 42-50.
- [6] Santana R A,Pontes M R,Bastos-Filho CJA.A multiple objective particle swarm optimization approach using crowding distance and roulette wheel[C].Ninth International Conference on Intelligent Systems Design and Applications, 2009: 237-242.
- [7] Deb K,Pratap A,Agarwal S,et al.A fast and elitist multi-objective genetic algorithms: NSGA II[J].IEEE Transactions on Evolutionary Computation,2002,6(2):182-197.
- [8] Kennedy J,Eberhart R. Particle swarm optimization [C]. Perth: Proceedings IEEE International Conference on Neural Networks,1995:1942-1948.
- [9] Zitzler E,Deb K,Thiele L.Comparison of multi-objective evolutionary algorithms: Empirical results[J].Evolutionary Computation, 2000, 8(2):173-195.
- [10] 徐刚,瞿金平.一种用于多目标优化的混合粒子群优化算法[J].计算机工程与应用,2008,44(33):18-21.
- [11] 李宁,邹彤,孙德宝,等.基于粒子群多目标优化算法[J].计算机工程与应用,2005,41(23):43-46.
- [12] 钱伟懿,李阿军,杨宁宁.基于混沌的多目标优化粒子群优化算法[J].计算机工程与设计,2008,29(18):4794-4796.